Interdependent Components for the Development of Accessible XUL Applications for Screen Reader Users

Xabier Valencia[1], Myriam Arrue[1], Halena Rojas-Valduciel[2] and Lourdes Moreno[2]
[1]*EGOKITUZ: Laboratory of HCI for Special Needs, University of the Basque Country (UPV/EHU), Informatika Fakultatea 20018, Donostia, Spain*
[2]*Computer Science deparment, Universidad Carlos III de Madrid, 28911, Leganés, Spain*

Keywords: Accessibility, User Testing, Expert Review, XUL, Mozilla Firefox Browser, Screen Reader.

Abstract: Web applications based on XUL technology have reached great development. This technology enables developers to easily create extensions and add-ons of Mozilla Firefox browser. It is essential to keep in mind accessibility in the development of such applications in order to not discriminate user groups. In this sense, standards and good practices have to be considered. Furthermore, User-Centred Design and Inclusive Design approaches should be followed as they involve users with disabilities in the development process. This paper presents an analysis of XUL accessibility guidelines created by Mozilla Foundation. An accessible XUL application has been designed and developed based on the guidelines. User testing has been conducted by two blind users revealing several important accessibility barriers. In addition, an expert review process was carried on by a blind accessibility consultant. They all used JAWS screen reader. The results obtained show that the existing guidelines conformance is not enough for ensuring accessibility of the application. There are other factors dependent on assistive technologies and user agent that have to be considered in the development of accessible XUL applications.

1 INTRODUCTION

The use of Internet has experienced a vertiginous growth in the last few years. Users access the Web employing diverse devices, modalities and technologies. Due to this diversity, inclusion approaches are necessary in order to provide full accessibility to Web contents and avoid the exclusion of some user groups.

Users with disabilities are the most affected by accessibility barriers on the Web. They access the Web using assistive technologies, for example, a screen reader that relates content in audio to the visually impaired. It is essential to develop accessible web applications to ensure appropriate assistive technology support.

Currently, research work regarding web browser functionality augmentation is gaining attention. Some examples of Mozilla Firefox add-ons are (Greasmonky, 2012), (Stylish, 2013) and (Turn off the Lights, 2013).

These augmented functionalities could be utilized by all users only if accessibility aspects are considered in their development process. Thus, a comprehensive inclusive design paradigm for augmented browser functionalities should integrate User-Centred Design (UCD) methods and Inclusive Design approaches in addition to accessibility guidelines compliance (Lawton, 2007) (Newell, 2000).

The interest of "design for all" paradigm is rapidly increasing in the community and several efforts have been made in this way. In fact, there are several organizations concerned with web accessibility. They develop and maintain support resources for complying with accessibility standards such as guidelines. This is the the case of Mozilla Foundation (XUL, 2013b). In addition, it provides extension mechanisms to augment browser functionality and develop application add-ons for Mozilla Firefox through one specific technology such as XUL (XUL, 2013a).

The objective of this paper is to analyse the appropriateness of the set of accessibility guidelines defined for XUL technology. For achieving this objective, an accessible add-on for augmenting Mozilla Firefox browser functionalities has been developed. In the development process, a

Valencia X., Arrue M., Rojas-Valduciel H. and Moreno L..
Interdependent Components for the Development of Accessible XUL Applications for Screen Reader Users.
DOI: 10.5220/0004855200650073
In *Proceedings of the 10th International Conference on Web Information Systems and Technologies* (WEBIST-2014), pages 65-73
ISBN: 978-989-758-024-6
Copyright © 2014 SCITEPRESS (Science and Technology Publications, Lda.)

comprehensive inclusive design paradigm has been applied, including UCD approach and user testing. User testing was performed by two screen reader users. Several accessibility barriers were observed in the testing which were later on evaluated by an expert screen reader user who works as an accessibility consultant. As a result, a review of the XUL accessibility guidelines and conclusions of the development process carried on are presented.

2 OVERVIEW

2.1 XUL

XUL (XML User Interface Language) is a XML based language to create User Interfaces (UIs), in the Mozilla platform. The main goal of the language is to allow easy development of cross-platform add-on applications which run on any Mozilla integrated platform.

It separates the program logic from the user interface components, facilitating the work of the designers and programmers. This approach is also applied in languages like (QML, 2013) or (XAML, 2013).

XUL is based on existing standards such as XML, HTML, CSS, DOM and Javascript. In addition, the Cross-Platform Component Object Model (XPCOM) technology can be applied (XPCOM, 2013) when operating system functionalities are required.

2.2 Related Work

In the development processes, technological, human and legislative aspects must be considered in order to manage accessibility issues. Consequently, related work from numerous disciplines should be taken into account. In the standardization field, the W3C ought to be highlighted along with the Web Accessibility Initiative (WAI) (WAI, 2013). The Web Content Accessibility Guidelines (WCAG) 2.0 (WCAG 2.0, 2008) is one of the most important components, and is viewed as the official standard.

The ISO 9241-210:2010 standard provides a framework for following and incorporating a UCD approach into a particular context of accessibility. Following methods that integrate usability and accessibility in products design processes will ensure that users with and without disabilities could be able to access. This is the distinguishing characteristic that User Sensitive Inclusive Design (Abascal et al., 2007) has; the user with disabilities is in mind. This work is focused on carrying out user testing technique in order to validate the accessibility included in a web application.

Several works related to XUL and accessibility has been found in the literature. These research works are related to developments of browser extensions for Mozilla Firefox. All of them are oriented to people with disabilities (Mirri et al., 2011) (Hanson et al., 2005).

Nevertheless, very few articles have been found that directly address the question of how to model accessibility according to the WCAG (Moreno et al, 2013), (Martin et al, 2010). An interesting attempt meriting particular mention is the Dante approach integrating the Web Authoring for Accessibility (WAfA) ontology (Yesilada et al, 2004) (Harper and Yesilada, 2007) for the visually impaired into WSDM (Plessers et al, 2005).

2.3 XUL Accessibility Guidelines

XUL language is based on web standards so its accessibility guidelines do not differ too much from previously published web accessibility guidelines.

The XUL accessibility guidelines are divided in six different sections: Keyboard Access, Assistive Information, Display, Human Computer Interaction, Media and Custom Widgets (XUL, 2013b). Each of one has a set of checkpoints to verify. For instance, the guideline Keyboard Access defines eight checkpoints that should be considered: one related to tab order, another one to keyboard shortcuts and so on.

All the guidelines and the related checkpoints can be seen as an accessibility checklist to be considered in order to evaluate the accessibility of a developed add-on application. The XUL accessibility guidelines document states a pass/fail statement to each checkpoint which could be applied in order to elaborate a checklist easy to verify by developers at design time. Table 1 presents the checklist for evaluating XUL accessibility guidelines.

3 EXPERIMENTAL DESIGN

3.1 Object of Study

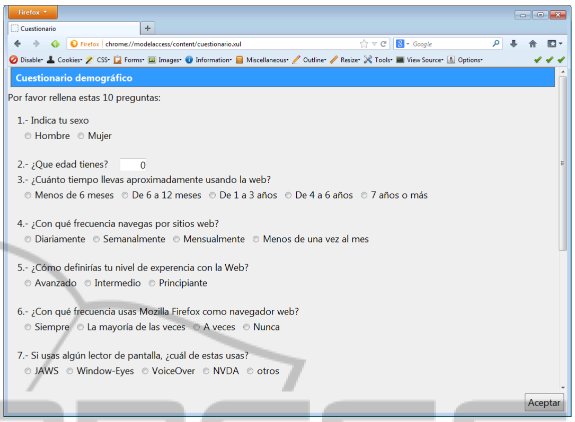
The object of study is to analyse the appropriateness of XUL Accessibility Guidelines for the development of accessible XUL-based applications.

Table 1: XUL accessibility checklist.

1.1	Logical tab order is provided
1.2	Keyboard functionality is provided for inaccessible features such as the column picker or added features such as column sorting
1.3	Keyboard alternatives are provided for toolbarbutton functionality
1.4	Keyboard shortcuts are present for important functionality
1.5	Context menus are triggered by the oncontextmenu event handler
1.6	All mouse operations have keyboard accessible equivalents
1.7	All scrollable elements are controllable with the keyboard
1.8	Keyboard focus is maintained and does not move unexpectedly
2.1	Alternative text is provided for meaningful images
2.2	All windows, including dialogs and wizards, have a descriptive title
2.3	Every form element has an associated label and radiobuttons are encapsulated in a groupbox
3.1	System settings are maintained
3.2	Color alone is not used to convey meaning and sufficient contrast exists between font color and background color
3.3	Visual elements and containers resize gracefully
4.1	Help documentation is provided including a description of keyboard shortcuts
4.2	Alerts are displayed using the alert scripting function or the notification box element
4.3	Interactive elements are sufficiently large and visible
4.4	Alerts are presented when the user initiates an error. The user has the opportunity and instruction to fix the error
4.5	User is informed of time limits and has control of response time when appropriate
5.1	Transcripts are provided for audio tracks
5.2	Video is captioned and a transcript is provided
5.3	User has control over animation and is warned about flashing content
6.1	Custom widgets provide accessible functionality

3.2 Experiment Context

Following XUL accessibility guidelines, a XUL-based application has been developed. Its accessibility has been evaluated using the XUL Accessibility Checklist (see Table 1). The developed application is an add-on for augmenting Mozilla Firefox browser functionalities.



Cuestionario demográfico

Por favor rellena estas 10 preguntas:

1.- Indica tu sexo
○ Hombre ○ Mujer

2.- ¿Que edad tienes? 0

3.- ¿Cuánto tiempo llevas aproximadamente usando la web?
○ Menos de 6 meses ○ De 6 a 12 meses ○ De 1 a 3 años ○ De 4 a 6 años ○ 7 años o más

4.- ¿Con qué frecuencia navegas por sitios web?
○ Diariamente ○ Semanalmente ○ Mensualmente ○ Menos de una vez al mes

5.- ¿Cómo definirías tu nivel de experencia con la Web?
○ Avanzado ○ Intermedio ○ Principiante

6.- ¿Con qué frecuencia usas Mozilla Firefox como navegador web?
○ Siempre ○ La mayoría de las veces ○ A veces ○ Nunca

7.- Si usas algún lector de pantalla, ¿cuál de estas usas?
○ JAWS ○ Window-Eyes ○ VoiceOver ○ NVDA ○ otros

Figure 1: Demographic information form.

3.3 Sample

The add-on application for Mozilla Firefox includes several pages with different type of web content.

For this study, two web pages containing forms have been selected. The selection of such type of application was due to the following reasons: the diversity of elements included in it, importance of accessibility in order to get to each question of the forms and fill it in and the high interaction it requires from users (Lazar et al., 2007).

The questions that users are required to fill in are about demographic information, emotional aspects and issues related to the design of the visited web pages. These forms are presented to the user after some specific time interval browsing in a website. Figure 1 shows one the forms developed based on XUL.

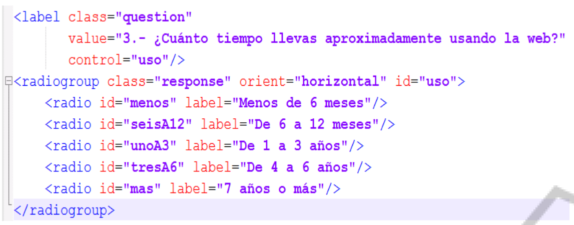
Forms were implemented using the following XUL elements: *Window, Radiogroup/Radio, Textbox, Label/Description, Image, Button, Hbox/Vbox/Box, DialogHeader and Spacer.*

The developed forms share a similar structure. We can resume this XUL structure in the following way:

- Each form is a *Window* element which has a *Box* as a container of the form.
- The title of the form is defined with a *DialogHeader* element.
- *Description* elements have been included for providing explanations.

The input elements are one of the following: text inputs, number inputs and radio inputs. In Figure 2 an extract of a XUL document is shown.

As it can be appreciated in Figure 2, a *label* has been attached to the *radiogroup* through the *control* attribute. This mechanism ensures that assistive technologies would adequately present the form to the user.



```
<label class="question"
       value="3.- ¿Cuánto tiempo llevas aproximadamente usando la web?"
       control="uso"/>
<radiogroup class="response" orient="horizontal" id="uso">
    <radio id="menos" label="Menos de 6 meses"/>
    <radio id="seisA12" label="De 6 a 12 meses"/>
    <radio id="unoA3" label="De 1 a 3 años"/>
    <radio id="tresA6" label="De 4 a 6 años"/>
    <radio id="mas" label="7 años o más"/>
</radiogroup>
```

Figure 2: XUL extract for demographic information form.

The other input elements have been implemented similarly, ensuring that all *title* values are unique and that all *labels* were attached to the corresponding *input* element

The implemented application was verified by developers using the checklist presented in Table 1. Not all the checkpoints are relevant to the developed application as some of them are considered content not usually present in common forms. The results obtained in this initial evaluation are presented in Table 2.

All checkpoints in the checklist were fulfilled with the exception of item 4.1, the help function. It was decided not to include this function in this preliminary version. However, for upcoming development iterations, the help function will be incorporated. In conclusion, we would not expect interaction problems with the application as almost checkpoints were fulfilled. Therefore, accessibility of the application was ensured.

3.4 Participants

Two screen reader users with more than 6 years of expertise browsing the web were recruited for the user testing: a woman (User 1) and a man (User 2) whose ages were 30 and 40 respectively. User 1 considered herself as an intermediate Web browser user whereas User 2 considered himself as an advanced user.

They both use JAWS, but one of them (User 1) uses it infrequently as she prefers to use VoiceOver screen reader on Mac OS operating system. User 2 uses Windows and JAWS, but he does not usually use Firefox. Even though, both are Mozilla Firefox sporadic users.

The experimental sessions were carried out in the same lab. They were asked to bring their laptop so they used JAWS configured with their personal preferences. The platform used was similar for both users: Windows operating system (User 1 used Windows XP and User 2 used Windows 7), JAWS 12 and Mozilla Firefox 22. Users were encouraged to report any barrier they detect when interacting with the XUL application. The sessions were recorded with a camera located behind the user in order to obtain information about the interaction. The interviews were taped with a voice recorder.

Table 2: The XUL accessibility checklist applied.

1.1	The tab order works correctly
1.4	Buttons have shortcuts
1.6	All actions are accessible from keyboard and mouse
1.7	The scroll can be done with the keyboard
1.8	The focus works as expected
2.1	The images have no alt text since are targeted to other users and it has nothing meaningful for blind
2.2	All windows have different titles
2.3	Labels are connected with their input element
3.1	Elements size has been set using “em” units
3.2	Elements colour have not meaning and the fonts and background has enough contrast
3.3	Flex elements has been used to avoid unexpected UI behaviours.
4.1	It is not provided in this draft version
4.2	Alerts are displayed using the alert scripting function
4.3	Form is clearly differentiated
4.4	Uncompleted form or errors are advised

In addition, an expert evaluation was performed by an expert screen reader user who has been working as ICT accessibility consultant at least for the last five years. All the evaluation was carried on at her usual working setting, and reported her findings by email.

3.5 Procedure

First, the XUL application was installed on users' laptop. Then, all users were asked to perform two tasks. The first one consisted on freely navigate during five minutes in a concrete website (www.discapnet.com). Then, the application presented the first form to complete. The questions in this form were related to their navigation experience. The second task consisted on a search task on the same website with a limited time interval of ten minutes. Finally, the application presented the second form containing questions related to demographic data.

4 RESULTS

4.1 User Testing

4.1.1 User 1

This user experienced several barriers when filling in the forms developed with XUL. The barriers reported by the user are the following. Besides, related XUL guidelines are indicated in each case:

- Barrier 1.1: She was unable to know which answer option was checked in the multiple-choice type questions. This occurred when the user wanted to verify that the selected answer was the correct one. She navigated with the virtual cursor of JAWS and it read the labels correctly. However, it only read the value of the option and informed that it was not checked even if it really was checked. Therefore, we had to tell her which option was selected in order to ensure that it was the desired one. (Related guideline: 2.3)
- Barrier 1.2: She experienced navigation problems. Surprisingly, when she filled in a multiple-choice question, JAWS focus was moved to the first question of the form. Then, she had to navigate to the next question from the beginning of the form. This happened even though the program focus was at the correct position. (Related guidelines: 1.1, 1.8)
- Barrier 1.3: JAWS shortcuts navigation feature did not work properly. Due to the difficulties she was having, she tried to navigate through the form using the JAWS shortcuts, like for instance, forms or headings shortcuts but it did not worked as expected. (Related guideline: 2.3)
- Barrier 1.4: Problems for clicking on a button. She was unable to find the button to continue. The button was located at the end of the form and was accessible using tab or arrows. However, JAWS did not correctly detect this element. (Related guideline: 2.3)

4.1.2 User 2

This user experienced similar problems reported by User 1 except of Barrier 1.3 (he did not try this mode of navigation). In addition, he reported other barriers:

- Barrier 2.1: Problems with text input questions. JAWS was unable to detect a text area element even if the focus was on one. Sometimes, he reported listening a label that was not the correct one. (Related guideline: 2.3)
- Barrier 2.2: Alert messages were not adequately presented. JAWS detected the alert windows but not the containing text. Therefore, he only could hear the default sound of the alert and "OK" button but he missed the alert message. (Related guideline: 4.2)
- Barrier 2.3: Problems with numeric type inputs. Firefox adds special controls for this type of inputs. These controls are for entering the numeric value using two small buttons inside the element, one of them for increasing the value and the other for reducing it. These controls entered into conflict with his navigation controls. They are activated with keyboard arrows which were the navigation mode used by this user. Consequently, he was unable to correctly enter his age. (Related guidelines: 1.6, 2.3)
- Barrier 2.4: JAWS read not existing options in the UI. He reported us that JAWS sometimes read text elements that were not in the UI. (Related guideline: 2.3)

4.1.3 Discussion of Results

The user testing carried on indicates that there are quite accessibility barriers in the developed XUL application, even if accessibility guidelines have been considered.

Some of the detected barriers are high impact ones as users could not complete the tasks without any assistance, for instance, barrier 1.4 (Problems for clicking on a button) experienced by both users. This barrier is related to activating the submit button of the forms. Users could not get to these buttons so they could not complete the tasks on their own. These types of barriers are accessibility problems, which should be documented in the accessibility guidelines. In the group of accessibility barriers should be also included the following ones: Barrier 1.1, Barrier 2.1, Barrier 2.2 and Barrier 2.3.

Other group of barriers detected in the user testing were of moderate impact, as they do not compromise the accessibility of the application. However, they make the application less usable and users can be disappointed inducing negatively in their accessibility perception. These barriers should be erased as well for ensuring a satisfactory user interaction. For instance, Barrier 1.3 (JAWS shortcuts navigation feature did not work properly) user has other alternatives of navigation with the screen reader so this problem does not compromise the correct completion of tasks. Even though, the inexistence of this barrier makes the application more usable and user experience could be more satisfactory. In the group of usability barriers should be also included the following ones: Barrier 1.2 and Barrier 2.4.

The detected barriers influenced negatively in the user satisfaction when interacting with the XUL application and they both needed around 15 minutes to fill in each form. Considering that each form consisted of ten short questions the time spent is not acceptable.

There are remarkable differences between the results obtained by each user. User 1 detected only two high impact barriers (Barrier 1.1 and Barrier 1.4) whereas User 2 experienced more barriers of this type (Barrier 1.1, Barrier 1.4, Barrier 2.1, Barrier 2.2 and Barrier 2.3). They both used the same version of the screen reader but the navigation strategies applied can differ a lot from user to user. In addition, our opinion is that User 1 is a more experienced user. Our observation in the experimental sessions revealed that User 1 has a wide range of knowledge about functionalities of screen readers. This observation differs from the perception of their own expertise as User 1 defines herself as an intermediate user and User 2 as an advanced one.

All in all, accessibility guidelines should consider all potential barriers independently of assistive technology version used, navigation strategies applied and user expertise level.

4.1.4 Improvements to XUL Guidelines

As can be seen most of the barriers are related to those guidelines regarding form elements, like the guideline 2.3 or the keyboard related issues 1.1 or 1.8.

Tagging labels with the corresponding control seems to be not enough. It is essential to correctly identify all questions and provide mechanisms in order to alert user and assistive technology about the existence of a list of questions or choices. Adding a new XUL element to tag the whole form would allow assistive technologies to handle better the information and also ensure the correct behaviour of the screen reader cursor.

Orientation of screen reader users would considerably improve by applying a simple good practice: informing at the beginning of the form about the total number of questions in it and numbering each question.

Regarding the keyboard, the added controls and shortcuts could create conflicts with browser controls or shortcuts as well as with assistive technologies controls. In this sense, information about the shortcuts available in the most used assistive technologies would be helpful. This would allow a better and more efficient navigation to the user and would avoid unexpected technology behaviour.

Finally, for the alert message issues, instead of using the standard alert element, the “notificationbox” should be used. In our preliminary tests, this element seemed to work better with JAWS. However, it may cause inconveniences to other type of users such as those using magnifiers. Another alternative solution could be to apply alert functions on the active window. This issue requires more investigation to carry on.

4.2 Expert Evaluation

Due to the distinct results obtained by the users and in order to obtain a factual knowledge of the situation; a review process was conducted by an expert blind screen reader user. She is a consultant on ICT accessibility.

She was asked to conduct a test with different versions of JAWS, Firefox and operating systems. A summary of the testing can be found in Table 3. Results of the testing show the strong dependency that XUL accessibility features have with the user agent, the operating system and version of the screen reader. For instance, checkpoint 2.2 “All windows have different titles” only can be correctly detected if the user interacts with the application on a Windows 7, JAWS 13 and Firefox 23 platform. The fulfilment of checkpoint 2.2 is not so essential for user interaction (the user could complete a task even if the title of windows are not correctly presented). However, the same platform is required for checkpoint 2.3 “Labels are connected with their input element”. This checkpoint is essential if the user is supposed to access and fill in questions presented in a form.

The expert noted that Firefox generates inaccessible HTML elements for the application interface. Therefore, JAWS does not correctly read them to the user. This is due to the way JavaScript implements and interprets the DOM. Visually, the element appears in the interface, but it is as if the element was non-existent in the DOM. The screen reader just ignores it.

In conclusion, the latest version of Firefox and JAWS (at least versions JAWS 13 and Firefox 23) seems to be the best combination for using the developed XUL application. Nevertheless, it is unusual that users have the latest versions of screen readers. The same occurs with browsers. Most of screen reader users use outdated browser versions since the cursor mode of JAWS does not work with the latest versions. Sometimes this mode is necessary in order to access user interface elements that cannot be read as usually.

Table 3: Correspondence of XUL accessibility checklist with the expert review results.

1.1	Only in Firefox 23.
1.4	Better using Firefox 23
1.6	Only using Firefox 23
1.7	Only using Firefox 23
1.8	JAWS 13 and Firefox 23
2.1	When running the extension on Windows 7, using Firefox 23
2.2	Windows 7, JAWS 13 and Firefox 23
2.3	Only using JAWS 13 and Firefox 23
3.1	Good feature for low vision using a magnifier
3.2	Good feature for low vision
3.3	Better using Firefox 23 and JAWS 13, JAWS 14
4.1	-
4.2	Better using Firefox 23
4.3	When refreshing the virtual buffer of JAWS, refresh the page or when the form is being read automatically
4.4	Only using Firefox 23 JAWS 13 and Windows 7

Some suggestions for improving the current version of the developed XUL application were indicated by the expert:

- The text for the questions should be defined as a header element.
- The multiple-choice questions should define the possible answers as a list element.
- Users should be provided by an input text element for introducing the answer. This suggestion would increase the form response time, therefore it would be less efficient and usable. However, it would be accessible for screen reader users.

Implementing these suggestions would overcome some of the most significant problems that users will have probably to face to. Mainly when they do not use the latest versions of browser and screen reader.

5 LESSONS LEARNED

From the results obtained in user testing and expert evaluation, it is clear that there are extremely important accessibility barriers in the developed XUL application. The user testing can bring up problems that have not been detected previously.

Findings reported in section 4 also highlight the importance of testing applications with different versions of Firefox and screen reader. There are too many differences between versions and possible combinations. Each combination has its strengths and weaknesses. Using the latest versions of each one seems to be the best solution. Even though, many people do not update their software, principally, due to the price of these updates. But sometimes there are some functionalities that the user is used to, that he cannot leave behind. For these reasons, it is crucial to ensure that the developed applications can be used in the wider range of versions as possible.

User agent developers should consider the accessibility barriers detected in this work. Universal access to existing augmented browser applications can be guaranteed only if inclusion design paradigms are adequately defined and applied.

In the mean time, a transitory solution could be to transform the application into a browser XUL element to display the forms coded in HTML inside the XUL code. As HTML accessibility issues have been more analysed and considered in the last decades. A great amount of documentation, tools and methods exist for making accessible HTML code. It would avoid the creation of the keyboard scripts, because the keyboard behaviour would work as expected and also reduces the workload. But the main advantage is that it would make possible a higher compatibility between different Firefox versions and screen readers.

Anyway, XUL accessibility guidelines should be reviewed in order to update issues regarding keyboard navigation, assistive technologies compatibility, forms elements tagging, etc. Not only to ensure that all elements are accessible but guarantee also the HTML-like behaviour so users do not get confused or disoriented during the interaction.

6 CONCLUSIONS

Web access to all users should be ensured, including people with disabilities who use assistive technology to access ICTs. Many organizations are concerned about this issue, and they work towards accessibility compliance. This is the case of the Mozilla Foundation that provides accessibility guidelines to apply when developers use their technologies like XUL.

This paper presents a study of XUL and its accessibility guidelines. We have developed an application for augmenting browser functionalities in XUL. The development process considered UCD approach with the aim of creating an accessible application with form content type. User testing with 2 legally blind users was carried on in addition to accessibility guidelines conformance evaluation. The developed application seemed to be accessible according to XUL accessibility guidelines. However, the results gathered in the user testing indicated important accessibility barriers. Some of the detected barriers make the application not operable for screen reader users.

An expert evaluation has also been considered in this paper. A screen reader user with more than five years of experience as a consultant on ICT accessibility has conducted the review. The results reveal a strong dependence between platform used (versions of user agent, operating system and screen reader) and the accessibility barriers experienced by users.

The findings of this paper should be considered in next versions of XUL accessibility guidelines. Some of them could be included as guidelines whereas others could be considered as best practices.

This research was oriented to screen reader users and XUL applications containing forms. In the near future, there is a need of performing evaluation studies with more content type, other groups of users and other assistive technologies. Therefore, future work will be motivated to the evaluation and improvement of other XUL accessibility checkpoints not considered in this work. This research work will lead to ensure universal access to Mozilla Firefox add-on applications.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge the support of the Spanish Ministry of Science and Innovation through Project ModelAccess (TIN2010-15549) and the MULTIMEDICA project (TIN2010-20644-C03-01).

EGOKITUZ is supported by the Department of Education, Universities and Research of the Basque Government (Eusko Jaurlaritza/Gobierno Vasco) through Grant# IT-395-10.

REFERENCES

Abascal, J and Azevedo, L. Fundamentals *of Inclusive HCI. Design*. (2007) Universal Access in Human Computer Interaction, 4th International Conference on Universal Access in Human-Computer Interaction, UAHCI 2007, Held as Part of HCI International 2007, Beijing, China, July 22-27.

Greasmonky (2012) http://wiki.greasespot.net/Main_Page.

Hanson, V. L., Brezin, J., Crayne, S., Keates, S., Kjeldsen, R., Richards, J. T., Swart, C., & Trewin, S. (2005). *Improving Web accessibility through an enhanced open-source browser.* IBM Systems Journal, 44 (3), 573 - 588.

Hanson V. L., Richards. J. T., and Swart. C., (2008) *Browser augmentation*, Harper S. and Yesilada Y, Web Accessibility, Springer London, pp. 215-229.

Harper, S. and Yesilada, Y., (2007). *Web Authoring for Accessibility (WAfA)*. Web Semantics: Science, Services and Agents on the World Wide Web. 5, 3, pp. 175-179.

Lawton S, H. (2007) *Just Ask: Integrating Accessibility Throughout Design. Madison,*: ET\Lawton, available at www.uiAccess.com/justask/

Lazar J, Allen A, Kleinman J, Malarkey C. (2007) *What Frustrates Screen Reader Users on the Web: A Study of 100 Blind Users***,** International Journal of human-computer interaction, Taylor & Francis, 22 (3), pp. 247-269.

Martín, A, Rossi, G, Cechich, A and Gordillo, S. (2010) *Engineering Accessible Web Applications. An Aspect-Oriented Approach.* World Wide Web, Springer US 13 (4), pp. 419-440,

Mirri, S, Salomoni, P, and Prandi, C. 2011. *Augment browsing and standard profiling for enhancing web accessibility.* In Proceedings of the International Cross-Disciplinary Conference on Web Accessibility. ACM, New York, NY, USA, Article 5 , 10 pages.

Moreno L, Valverde F, Martínez P, Pastor O, (2013). *Supporting accessibility in Web engineering methods: a methodological approach*, January, 2013, Journal of Web Engineering , RINTON PRESS, INC, ISSN: 12 (3&4), pp. 1540-9589.

Newell, A.F.; Gregor, P. (2000) *User Sensitive Inclusive Design: in search of a new paradigm*. Proceedings on the 2000 conference on Universal Usability, pp. 39-44.

Plessers P, Casteleyn S, Yesilada Y, De Troyer O, Stevens R, Harper S, and Goble C (2005) *Accessibility: A Web Engineering Approach.* In Proceedings of the 14th International Conference on World Wide Web (WWW '05) ACM, New York, NY, USA, pp. 353-362.

QML (2013) http://qt-project.org/doc/qt-5.0/qtquick/qtquick-applicationdevelopers.html.

Stylish http://userstyles.org/

Turn off the Lights (2013) http://www.stefanvd.net/project/turnoffthelights.htm.

WAI (2013) http://www.w3.org/WAI/

WCAG 2.0 (2008). http://www.w3.org/TR/WCAG20/

XAML (2013) http://msdn.microsoft.com/en-us/library/ms752059.aspx.

XPCOM (2013) https://developer.mozilla.org/en-US/docs/XPCOM.

XUL (2013a) https://developer.mozilla.org/en-US/docs/XUL.

XUL Accessibility Guidelines (2013b) https://developer.mozilla.org/en-US /docs/ XUL_accessibility_guidelines.

Yesilada,Y., Harper, S., Goble, C. and Stevens. (2004) R. *Dante annotation and transformation of web pages for visually impaired users*. In The Thirteenth International World Wide Web Conference. ACM, New York, NY, USA, pp. 490-491.