

© Springer-Verlag Berlin Heidelberg, 2013

This document is published in: Rocío Calvo, Ana Iglesias, Lourdes Moreno, (2013). An Accessible Chat Prototype for Screen Reader Users in Mobile Devices. HCI International 2013 - Posters' Extended Abstracts: International Conference, HCI International 2013, Las Vegas, NV, USA, July 21-26, 2013, Proceedings, Part I. Springer (Communications in Computer and Information Science, Vol. 373), pp. 216-220.

DOI: http://dx.doi.org/10.1007/978-3-642-39473-7_44

An Accessible Chat Prototype for Screen Reader Users in Mobile Devices

Rocío Calvo ¹, Ana Iglesias ¹, Lourdes Moreno ¹

¹ Computer Department, Universidad Carlos III, Leganés, Spain
{mrcalvo, aiglesia, lmoreno}@inf.uc3m.es

Abstract. Chats present accessibility problems for screen reader users. This work presents a prototype of an accessible chat for Mobile Devices (MD). The main aim of this research is to remove the accessibility barriers that screen reader users face when they use a chat in a MD. Thus, this prototype is based on the requirements which have been elicited considering background research and with the use of Software Engineering (SE) and Human Computer Interaction (HCI) methods as well as accessibility standards and guidelines.

Keywords: Accessibility, chat, interaction, screen reader, mobile device.

1 Introduction

The use of chats in Mobile Devices (MDs) is being increased in the last years. However, chats are not accessible for everybody [1]. For instance, screen reader users face accessibility barriers related to updated content without advising [2] and keyboard traps [3]. Furthermore, chats in MDs present additional accessibility barriers [4]. Related works, which include accessibility improvements, have been found; nevertheless, most of them do not follow a Use Centered Design (UCD) approach and do not include accessibility requirements to improve the interaction. Considering it, this research aims to create an accessible chat prototype which solves the barriers that screen reader users face when they use chats in MDs. To achieve it, some new or improved requirements have been added to progress the accessibility of chats in MDs.

2 Background

This section introduces the accessibility barriers that screen reader users face when use chats in MDs. Moreover, some previous accessible chats are studied.

2.1 Accessibility Barriers of Chats

Chats are synchronous communication tools and most of them present problems of accessibility for a huge variety of people. However, this kind of tools in MDs could present additional problems than other information technology (IT) systems like in-

teraction problems or auto-refresh content [5]. Furthermore, some people cannot follow the conversation [6]. For instance, if one of the emitters is not able to write quickly and if there are not ways to establish the turn to write, he could have problems to follow the rhythm of the conversation [7][8]. Besides, screen reader users face other specific problems related to the web page's auto-refresh or updated content because it causes the restart of the screen reader [3]; the use of keyboard traps [4]; or the lack of support for text-to-speech or text-to-braille functions [5].

With regard to the accessibility problems of chats in MDs, these problems are similar to the chat's barriers of desktop computers. However, there are some additional specific problems that users could experience due to the technology of MDs and nevertheless of their functional limitations [2].

2.2 Previous Proposals of Accessible Chats.

Some researchers have tried to solve the accessibility problems of chats. For instance, the Reef Chat [9] follows the specifications of WCAG 2.0 [10] and WAI-ARIA [11]. Moreover, the mobile chat, AssistiveChat¹, provides features for people with speech disabilities like the suggestion of words.

Furthermore, some learning software like Moodle or Atutor, which include chats, has improved their chats. For instance, Moodle avoids the use of frames or Javascript technology and the auto-refreshing time can be configured [12]. Also, Atutor's Chat solves some technological aspects to guarantee access to users who use assistive technology because they can configure the refresh rate of the application. These approximations try to improve the user experience. However, none of them follow the UCD approach and most of them do not accomplish with accessibility guidelines. Moreover, they do not solve accessibility problems related to the interaction and the flow and rhythm of the conversation. Thus, the presented research aims to solve the interaction problems following a UCD approach and basing on standards and guidelines

3 Accessible Chat Approach for Mobile Devices

This work is part of a Ph.D which main goal is to create a model-driven design and a strategy development process of an accessible chat [13]. Currently, the Ph.D is in the requirement engineering phase. The requirements have been elicited and now they must be validated. To achieve it, the prototype has been created. This section details the elicited requirements and the chat prototype.

3.1 Elicited Requirements

The requirements elicitation phase combines SE and HCI methods as a complement of each other to include the user in this phase and follow a UCD approach. Some of these techniques are: brainstorming, user interviews, analysis of existing

¹ <http://www.assistiveapps.com>

system/competitors, scenarios and personas and standards and guidelines related to accessibility [14]. The obtained functional requirements which improve the user experience and accessibility in MDs are showed in the **Table 1**. It is important to emphasize, that some of these requirements are new functionalities and other requirements are improved classical functionalities of chats.

Table 1. Functional Requirements to Improve the User Experience and Accessibility in MDs

Requirement	Description of the Improvement
Add an interlocutor	Users could stop the new interlocutor addition to the conversation.
Predefined Sentences	Users can select predefined sentences provided by the system.
Add File	The user should specify a description for the uploaded file and the system informs the users about the size's file.
Add URL	The user should specify a summary of the URL and its language.
Refresh Conversation	Allow users to pause and refresh the conversation.
Convert Conversation	Transform the conversation to other formats like audio or braille.
Messages Preferences	Include preferences related to the way in which messages are showed such as: time refresh interval or chronological order.
Clean Messages	Allow users to clean the messages which are showed in the screen.
Reception Messages	Inform users when the message has been delivered.
Writing	Inform users when other users are writing.
Check Spelling	Inform users about grammatical errors.
Translate	Translate messages if the sentence language is different to the predefined language.

3.2 Chat Prototype

Basing on the obtained requirements the chat prototype is created to improve the accessibility and user experience in MDs. Next sections explain the chat development.

Responsive Web Design.

The prototype has been created to be adapted to different viewports. It has been developed in a web environment because it can be executed in more MDs than native applications and a responsive web design with the use of standards like HTML5 and CSS3 is followed to guarantee the layout adaptation to a huge variety of MDs [15].

Chat Functional Requirements to Improve Accessibility and Interaction.

The prototype includes some of the functional requirements which improve the accessibility and the interaction of the chat in MDs. The selected requirements for the prototype are: the *Refresh Conversation* and the *Add File*. It is important to remark that these requirements are improved using the WAI-ARIA specification because it adds semantic information to HTML code to specify screen readers how to align keyboard navigation to landmarks, the page structure, updated content and expanded information[11]. Next, these requirements are explained in detail.

With regard to the *Refresh Conversation*, the user is able to control the flow and the rhythm of the conversation. If the user feels overwhelmed, he could pause the conversation by pressing the button *Pause*. The system informs the other users about it, they could write more sentences but the user who has stopped the conversation does not receive any sentence until he refreshes the conversation again. Then, these messages will be showed together. The **Fig. 1** includes screenshots of the chat related with the “pause/refresh” functionality. The first figure shows the perspective of the user who stops the chat and the second one the perspective of the other user.



Fig. 1. Refresh/Pause Functionality

Moreover, the *Add File* requirement is improved too, **Fig. 2**. Users exchange files between them but sometimes the user cannot access to them. Thus, if the user sends a file, he should specify a description of the file and the system should inform the other users about the size and the description of the file. Then, the user who receives the file decides if he downloads it or not depending on his network circumstances and his necessities.

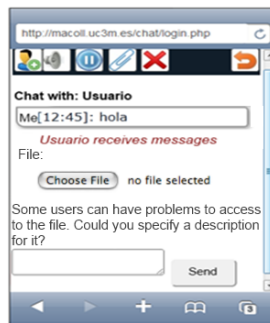


Fig. 2. Add File Functionality

4 Conclusion

This paper reveals accessibility barriers that screen-reader users face when they interact with chats. Previous works include accessibility improvements in this kind of tools. However, most of them do not include accessibility improvements in the user-interaction. The chat prototype presented in this paper has been designed following a

UCD approach including specific features for improving the screen-reader users experience like the *Refresh Conversation* or *Add File* which have been created using WAI-ARIA specification.

Currently, a user evaluation of the prototype is being carried out to check the accessibility of the tool and the effectiveness of the new proposed features.

5 Acknowledgements

This research work has been partially supported by MA2VICMR (S2009/TIC-1542) and MULTIMEDICA (TIN2010-20644-C03-01).

References

1. Hackett, S., Parmanto, B. and Zeng, X. Accessibility of Internet websites through time. In: the 6th International ACM SIGACCESS Conference on Computers and Accessibility, pp. 32-39, ACM Press, New York, (2004).
2. W3C. Shared Web Experiences: Barriers Common to Mobile Device Users and People with Disabilities, <http://www.w3.org/WAI/mobile/experiences>
3. Lazar, J., Allen, A., Kleinman, J. And Malarkey C. What Frustrates Screen Reader Users on the Web:A Study of 100 Blind Users. In. International Journal of Human-Computer Interaction Vol. 22(3) pp.247–269, Taylor and Francis (2007)
4. Resta, P. and Laferrière, T. Technology in Support of Collaborative Learning. In. (Eds.) Education Psychology Review Vol.19 (1) pp. 65-83, Springer (2007)
5. Schoeberlein, J. and Wang, Y. Evaluating Groupware Accessibility. In. 5th International Conference UAHCI pp. 414–423, Springer, Heidelberg (2009).
6. IMS Access For All v 2.0. <http://www.imsglobal.org/accessibility/>
7. National Center on Accessible Information Technology in Education. Accessibility of Electronic Tools & Features Used in Distance Learning, http://adasoutheast.org/ed/edpublications/itseries/8_etools.pdf
8. Guenaga, M., Burguer, D. and Oliver, J. Accessibility for e-learning environments. In: K. Miesenberger et al. (Eds.) Computers Helping People with Special Needs, LNCS, vol 3118, pp. 157-163, Springer , Heidelberg (2004)
9. Thiessen P and Russell E. WAI-ARIA live regions and channels: ReefChat as a case example. In: Disability and Rehabilitation. Assistive Technology. Vol. 4(4) pp. 276-287, Informa Allied Health (2009)
10. W3C. Web Content Accessibility Guidelines, <http://www.w3.org/TR/WCAG/>
11. W3C. Accessible Rich Internet Applications (WAI-ARIA) 1.0. <http://www.w3.org/TR/wai-aria/>
12. Moodle. Using chat. <http://docs.moodle.org/23/en/Chat>
13. Calvo, R., Iglesias, A. and Moreno, L. Accessible Chats for Computer Supported Collaborative Learning Environments in Mobile Devices (Doctoral Consortium). In: 7th IEEE International Conference on Research Challenges in Information Science. In press. (2013)
14. Calvo, R., Moreno, L. and Iglesias, A. Requirements elicitation for designing an accessible chat In: Fernández, L. (Eds.) REICIS Spanish Journal on Software Engineering, Quality and Innovation vol. 8(1) pp. 7-21 (2012)
15. Gardner, B. Responsive Web Design: Enriching the User Experience. In: Gilbert, H., (Eds.) Sigma Journal: Inside the Digital Ecosystem. Vol. 11(1) pp. 13-20. (2011)