

Making Wikipedia Editing Easier for the Blind

M.Claudia Buzzi, Marina Buzzi

IIT-National Research Council
via Moruzzi 1
I-56124 Pisa, Italy
+39-050-3152632/1

{Claudia,
Marina}.Buzzi@iit.cnr.it

Barbara Leporini

ISTI-National Research Council
via Moruzzi 1
I-56124 Pisa, Italy
+39-050-3152034

Barbara.Leporini@isti.cnr.it

Caterina Senette

IIT-National Research Council
via Moruzzi 1
I-56124 Pisa, Italy
+39-050-3152195

Caterina.Senette@iit.cnr.it

ABSTRACT

A key feature of Web 2.0 is the possibility of sharing, creating and editing on-line content. This approach is increasingly used in learning environments to favor interaction and cooperation among students. These functions should be accessible as well as easy to use for all participants. Unfortunately accessibility and usability issues still exist for Web 2.0-based applications. For instance, Wikipedia presents many difficulties for the blind. In this paper we discuss a possible solution for simplifying the Wikipedia editing page when interacting via screen reader. Building an editing interface that conforms to W3C ARIA (Accessible Rich Internet Applications) recommendations would overcome accessibility and usability problems that prevent blind users from actively contributing to Wikipedia.

Categories and Subject Descriptors

H.5.2 [Information Interfaces and Presentation]: User Interfaces – *Graphical user interfaces (GUI)*. K.4.2 [Social Issues]: Assistive technologies for persons with disabilities.

General Terms

Design, Human Factors.

Keywords

Wikipedia, accessibility, usability, blind.

1. INTRODUCTION

Wikipedia is the best-known Free Encyclopedia available on-line, based on MediaWiki free SW. Wikipedia's articles are written collaboratively by volunteers from all around the world, and the vast majority of its articles can be edited by anyone with access to the Internet [8]. Collaborative environments allow multiple users to share content creation. Thus simplicity is crucial for universal participation.

For the blind, who cannot use screen and mouse, and are obliged to interact through assistive technologies (i.e. screen reader and

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. NordiCHI 2008: Using Bridges, 18-22 October, Lund, Sweden Copyright 2008 ACM ISBN 978-1-59593-704-9. \$5.00.

voice synthesizer) the interaction is much more complex. In a previous study [2] we discussed difficulties encountered by the blind when interacting with Wikipedia via screen reader. In this paper we focus on the editing interface that presents the most difficulty for the blind, and propose an ARIA-based solution to facilitate editing. In the following we describe changes to the HTML source file of Wikipedia editing user interface to improve accessibility and usability via screen reader, and briefly discuss our choices as well as related open issues.

2. ACCESSIBILITY AND USABILITY ISSUES

When designing interfaces it is crucial to remember that interaction via screen reader is particularly complex since in addition to content this assistive technology should also communicate to the user the various UI components (links, buttons, etc.) and any information regarding events and changes (update of interface portions, change of focus, etc.).

Here we only consider interaction with Jaws (<http://www.freedomscientific.com>) since it is the most widely used screen reader in Italy [1]. Other screen readers may offer different commands but the basic functions discussed here are analogous.

The Wikipedia editing page is particularly problematic and poses three main usability issues for users who operate via keyboard and voice synthesizer [2]:

1. Depending on the navigation style used by the blind person, it is difficult (if not impossible) to apply formatting and style properties (as well as inserting links and images).

A navigation strategy frequently used by the blind is navigating by *Tab* Key [1]. Unfortunately, the formatting widgets of Wikipedia are inaccessible via *Tab* key, i.e. they do not receive the focus, since they are graphic icons, generated by JavaScript (Fig. 1), usable only with a mouse.

Another frequently used navigation strategy is sequential exploration via arrow keys. In the exploration modality of Jaws (for web navigation) the widgets are announced by the screen reader since an alternative description is assigned to the toolbar icons (i.e. "graphic bold clickable"). However the user must understand that each icon is associated with a function and should know how to use it. This is difficult for unskilled users.

An additional method for finding a specific widget, which requires the user to be familiar with the UI, is to use the

“find” command on the page (for instance, “find: Italics”). However, is not easy to understand that control elements may be found in the page as a text search.

Other advanced Jaws commands (for instance “g” for jumping from image to image) require considerable user effort and most users who only use basic commands are not even aware of this possibility.

Once a widget is found and a formatting option is selected by arrow key, the corresponding function can be activated by pressing the space bar. This works in IE, but unfortunately at this time it does not run in Firefox.

2. It is difficult to select special characters and symbols.

To insert a special character or symbol, a list of over five hundred links is available, unusable for those who navigate either via *Tab* key or by the list of all the available links (Insert+F7 Jaws command).

Furthermore, Jaws does not recognize uncommon symbols or characters, so it announces text that is hard to understand (for instance é, è, Ê and Ê are all announced as “link e”). A more descriptive text should be associated (e.g. “e” with acute accent) for clarity (Fig. 5).

3. It is difficult for blind users to switch back and forth between editing and selecting modalities.

Using mouse and keyboard to point, select and edit allows the user to operate easily. A character, symbol or graphical style property can be selected via mouse, while typing the content via keyboard since visually it is easy to control and move the edit cursor. Instead, a blind person performs both actions via keyboard, so interaction should be easier.

In the original Wikipedia editing page, to format a portion of text the user must switch between editing and the navigation modalities as follows:

- Presses *Enter* key to activate the *editing mode*
- Types text and selects the word to be formatted
- Presses the + key (numpad) to activate the *exploration mode*
- Selects desired widget via arrow keys or find command
- Presses *Enter* key (to apply formatting) and the focus returns to text box
- Presses *Enter* key to activate the *editing mode* (to continue text entry).

In the Wikipedia editing UI the focus is managed via JavaScript: when one or more words in the text area are selected all related parameters (including the focus) are stored by the script in order to apply the formatting correctly. However, a blind user cannot understand how the focus is processed since a “virtual focus” is available when interacting via screen reader [1]. This focus may not coincide with the system focus.

To overcome these difficulties, we have redesigned the Wikipedia editing page by using the Accessible Rich Internet Applications Suite (WAI-ARIA) [9]. ARIA aims to make dynamic web content

and applications more accessible to people with disabilities. It mainly focuses on dynamic content and advanced user interface controls developed with Ajax (asynchronous JavaScript and XML) [3], (X)HTML, JavaScript, and related technologies [9].

With ARIA it is possible to mark regions and define sections of a web UI (such as menus, main contents, sponsored links), and to enable users to rapidly move between regions/sections via keyboard, instead of pressing the *Tab* key many times. For instance, in order to explore this technology’s potential [6], authors built an accessible chat system using the ARIA suite. Other very useful examples of ARIA code were developed by the Mozilla Developer Center and the Illinois Center for Information

```
<div id='toolbar'>
<script type='text/javascript'>
/*<![CDATA[*/
addButton("/skins-1.5/common/images/button_bold.png","Bold
text","\\"","\\"","Bold text","mw-editbutton-bold");
addButton("/skins-1.5/common/images/button_italic.png","Italic
text","\\"","\\"","Italic text","mw-editbutton-italic");
...
</script>
</div>
```

Technology Accessibility (University of Illinois) [4], [7].

Figure 1 –Wikipedia editing page, Javascript toolbar.

3. REDESIGNING THE EDITING PAGE

We wish to show how complex, appealing UIs can become usable for the blind if a few basic issues are acknowledged and some key points are considered in the design phase. To fully explore the potential of the ARIA specification, our design/implementation work was divided into two phases: first we developed the new editing page using only XHTML, CSS and JavaScript; later we added ARIA properties, roles and methods. In this way it was possible to evaluate the benefits obtained by ARIA technology. We observed that in our case the use of ARIA allowed further enhancement of interface usability; indeed the first solution implemented without ARIA is already accessible but still complex. The second one--developed with ARIA properties--makes interaction easier and more comfortable for users who navigate by keyboard. ARIA resolves usability issues regarding the toolbar and the focus (see 3.1 and 3.3).

We tested our developed interfaces with MS IE v. 7.0, Mozilla Firefox v. 3.0 and the screen reader Jaws v. 9.0. The new UIs were tested by this paper’s authors independently; afterwards, outcomes were compared and discussed. One author has been totally blind since childhood, has extensive knowledge of the Jaws screen reader, and uses advanced commands daily. Sighted authors carried out the test using basic Jaws commands, interacting via keyboards, with PC monitor both on and off. The different experiences of the authors when using the screen reader allowed us to cover a variety of interaction modalities, leading to a more accurate test (i.e. basic commands, which simulate the use of beginner users; advanced screen reader functions, which are used by skilled individuals).

3.1 Formatting toolbar

As described above, the first issue we considered concerns the Wikipedia formatting toolbar, which is created with JavaScript.

This makes widgets practically inaccessible for a blind person who navigates via *Tab* key, since an image is not a control element, and the screen reader is unable to recognize it as a button. This problem can be resolved by using standard XHTML input elements with associated images (i.e. buttons), while maintaining the page's graphical appearance and providing accessible widgets. In our interface we used this solution. Each widget is associated with a specific access key (i.e. Ctrl+b for Bold, Ctrl+i for Italics, etc.) and with an appropriate value of the *Tabindex* attribute (Fig. 2). This simple change alone makes the toolbar accessible via screen reader.

However, since 21 widgets are present in the interface, using access keys requires users to memorize the shortcuts, so cognitive effort is also required, considering that browsers and screen readers (including Jaws) provide numerous shortcuts.

```
<div id="toolbar">
<div id="mytoolbar" class="hide"> formatting widgets</div>
<input type="image" Tabindex="2" accesskey="b" name="bold
button" onclick="return buttonEvent(event);" title="bold button"
style="cursor: pointer;" alt="Bold" src="button_bold.png" id="mw-
editbutton-bold" class="mw-toolbar-editbutton" border="0"
height="22" width="23">
</input>
```

Figure 2. Wikipedia editing page, XHTML toolbar.

Alternatively, the *Tab* key must be pressed many times to navigate the entire toolbar; in this sense the UI is accessible, but navigation is quite long and boring. Thus to improve interface usability, we applied ARIA specification to the XHTML code, making interaction easier and more comfortable. ARIA adds semantic information that communicates the object role (for instance role = "button") to the user. In this way it is also possible to use a graphic icon as a button, with the advantage that the screen reader can recognize it as a control element.

To create the formatting toolbar using ARIA, we defined roles and properties as shown in Figure 3. We used the *activedescendant* attribute in order to make the toolbar navigable via arrow keys. Once the toolbar receives the focus via *Tab* key, the child elements -- i.e. each formatting widget -- can be accessed by up and down arrows, and can be activated by pressing the ENTER key, which applies the associated formatting function (Bold, Italics, etc.). In this way the virtual focus is never lost during the editing process. This also resolves problem 3, described in more detail in the following.

```
[...]
<div role="listbox" Tabindex="2" id="listbox1"
onclick="return listitemClick(event);"
onkeydown="return listItemKeyEvent(event);"
onkeypress="return listItemKeyEvent(event);"
aria-activedescendant="listbox1-1"
aria-labelledby="mytoolbar">
<div id="mytoolbar" class="hide"> formatting buttons</div>
<span role="button" aria-pressed="false" class="selected"
id="listbox1-1" onclick="return buttonEvent0(event);"
title="bold button">

</span>
```

Figure 3. XHTML code of the new Wikipedia editing page.

Table 1 shows the order in which elements of the new Wikipedia editing interface receive the focus via *Tab* key. In this way, the sequence in which elements receive the focus reflects the logic pathway the user performs when operating with the interface. This visiting order permits accessing the editing fields before the formatting toolbar, as usually occurs.

1	Edit box
2	Edit summary
3	Toolbar
4	Combo-boxes
5	Push buttons

Table 1. Visiting order using *Tab* key

3.2 Special characters and symbols

The second usability problem we considered is related to special characters and symbols. In the new UI we arranged the 500 links in two different combo boxes: the first allows users to choose a 'language' and the second contains all the characters of the selected alphabet (Fig. 4).



Figure 4. The new Wikipedia editing page.

This more compact solution greatly facilitates interaction for the blind since the user may apply a special character in a selected alphabet without being forced to listen to all the previous characters (at worst, 500 symbols!). Figure 5 reports a fragment of the content announced by Jaws in the original version (on the left) and in the new one on the right.

...	...	To insert a symbol select it and press enter:
à	Link a	Combobox Select a group (1 of 3)
À	Link a	Latin
...	...	Wiki
è	Link e	Mathematics
È	Link e	
...	...	Combobox Select a character
ì	Link i	A with grave accent
Ì	Link i	Capitalized A with grave accent
...

Figure 5. The Wikipedia editing page read by Jaws: the original (center) and the new version (right).

This makes the content accessible for the blind but also impacts on sighted users since the GUI is modified; however, on the whole we believe that its usability is improved for everyone.

Furthermore, to simplify selection of the desired character, we specified a *label* attribute for each <option> item of the second

combo-box, so the screen reader can announce a description of the special character selected.

3.3 Focus

The third problem, regarding the focus, is completely resolved by our new ARIA-based code. The new interface allows users to insert and edit text without having to switch to navigation modality to find the interface elements (widgets and comboboxes). In short, the user activates the Editing modality and it remains for the entire editing/formatting process.

Specifically, when the user accesses the text box he/she:

- presses the *Enter* key to activate the *editing modality*
- types text and selects the words to be formatted
- presses the *Tab* key once to access the toolbar and selects the desired widget via arrow keys
- presses the *Enter* key (to apply the formatting property) and the focus returns to the text box to continue the text input.

Analogously, to insert a special symbol the user:

- presses the *Tab* key to access the first combo box
- presses the *Enter* key to activate the *editing modality*
- selects an alphabet via arrow keys
- with another *Tab* key press, accesses the second combo box and selects the desired symbol via arrow keys
- presses the *Enter* key so the symbol is inserted and the focus returns to the text box to continue the text input.

The ARIA *activedescendant* attribute (associated with the toolbar) allowed us to exploit the Wikipedia Javascript, which once the *keypress* event of the formatting button is captured, applies the related function and moves the focus back to the text box. In the original UI, this Javascript was only called by the mouse click.

In our first XHTML-based version (without ARIA) Jaws detects only a sequence of images, without announcing that each image is indeed a button (it announces: “graphic bold *onclick* event”). An expert user could figure out the association with the formatting function. In fact, by pressing the space bar the bold function is applied but note that the focus does not come back to the text box since Jaws is in *exploration modality*.

4. DISCUSSION

In order to minimize impact on sighted users, we tried to limit changes in the graphical UI of the new Wikipedia editing page; however, we accepted the introduction of the two combo-boxes since we believe that this may simplify interaction for all.

Different browsers may have different rendering and behaviors. Starting from version 7, Jaws performs well with both Mozilla Firefox and MS IE, so we tested our interface with both browsers. The new Wikipedia editing page performs well with Mozilla Firefox v. 3.0 while, at the moment, IE 7.0 does not interpret ARIA specification, which will be included in IE 8 [5].

With IE we also encountered a problem when selecting special symbols. When exploring the second combo box to select a character, *all* the symbols navigated by arrow key are inserted, not just the final desired symbol (i.e. the focus activates the selection). This problem is not present in Firefox v.2 and higher, which works correctly.

Lastly, note that shortcuts defined in `` elements in the ARIA specification do not run since they only apply to standard control elements.

5. CONCLUSION

With the expansion of web 2.0, everyone should be able to participate in social networking, regardless of physical ability. Wikipedia like other Wikis is a great opportunity for integrating the blind in today’s Internet, if interaction is properly designed and content delivered quickly and easily.

In this paper we have proposed some simple updates of the Wikipedia editing page, which greatly improve accessibility and usability via screen reader. Our study shows that the ARIA suite improves usability considerably. Accessibility can also be achieved using standard (X)HTML elements.

In the future we plan to carry out a usability test with totally blind users to compare their interaction with the original Wikipedia editing page and the new ARIA-based UIs discussed in this paper. We also intend to extend this research to blogging and social network sites.

6. REFERENCES

- [1] Patrizia Andronico, Marina Buzzi, Carlos Castillo, Barbara Leporini. 2006. Improving search engine interfaces for blind users: a case study. *Universal Access in the Information Society*, Vol. 5, 1 (June 2006), Springer, 23-40. DOI= 10.1007/s10209-006-0022-3
- [2] Marina Buzzi, Barbara Leporini. 2008. Is Wikipedia usable for the blind? In *Proceedings of W4A 2008 (Beijing, China, April 2008)*, ACM Press, New York, NY, 15-22.
- [3] Mozilla Developer Center. AJAX. <http://developer.mozilla.org/en/docs/AJAX>
- [4] Mozilla Developer Center – ARIA. http://developer.mozilla.org/en/docs/Accessible_DHTML
- [5] Microsoft. Internet Explorer 8 Readiness Toolkit <http://www.microsoft.com/windows/internet-explorer/beta/readiness/developers-new.aspx#w3c>
- [6] Thiessen, P., Chen, C. Ajax Live Regions: ReefChat Using the Fire Vox Screen Reader as a Case Example. In *Proceedings of W4A07, (Banff, Canada, May 2007)*, ACM Press, New York, NY, 136-137.
- [7] Illinois Center for Information Technology Accessibility <http://test.cita.uiuc.edu/aria/>
- [8] Wikipedia. Wiki, <http://en.Wikipedia.org/wiki/Wiki>.
- [9] W3C. WAI-ARIA Overview, <http://www.w3.org/WAI/intro/aria.php>.