Increasing Usability of Search Engine Interfaces for the Blind

Patrizia Andronico, Marina Buzzi

IIT-CNR
Via G. Moruzzi, 1 – IT-56124 PISA
{Patrizia.Andronico, Marina.Buzzi}@iit.cnr.it

Barbara Leporini[*]

ISTI-CNR
Via G. Moruzzi, 1 – IT-56124 PISA
Barbara.Leporini@isti.cnr.it

Abstract

Since search engines have become an indispensable tool for retrieving information from the Internet, simple and effective interaction is important for anyone wishing to explore a web page. Navigating is especially difficult for those blind individuals who navigate via screen reader. We believe that an appropriately designed user interface is crucial for improving accessibility and usability, and in a previous phase of our research we proposed a set of specific guidelines that would improve interaction with search engine user interfaces.

In this paper we applied our guidelines to a commercial search engine, modifying the Google interface source code while maintaining the same graphical layout. For a better understanding of how screen readers read a web page and how our changes improve readability via assistive technology, we offer two scenarios in which two blind persons were assigned the same task. We then show the two individuals attempting to perform the task, one with the original and the other with the modified Google interface, and briefly compare the two scenarios.

1 Introduction

This paper is a development of our ongoing in-depth study on search engine user interface accessibility and usability (Buzzi, Andronico & Leporini, 2004). The search engine user interface is composed of many features such as "arrangement of components", "expressive power", "number of elements", "result clustering" and so on. Usually, all these features are only considered and improved to facilitate visual interaction; consequently, when navigating by means of a screen reader it can be quite difficult to explore and execute a task.

The screen reader deals with Web page content in a manner very different from visual rendering. Interaction requires a certain expertise in advanced screen reader and browser commands, and orienting oneself within the page contents can require considerable effort. However, we will show that user interaction is greatly improved and simplified if the contents of search engine user interfaces are appropriately grouped according to function/logical categories. Our focus is on aural perception vs visual experience.

Considering the many difficulties that accompany screen reader interaction, we have proposed (Leporini, Andronico & Buzzi, 2004) the following eight guidelines for improved search engine UIs, in order to simplify navigation and the perception of the whole page content:

1. *Easy location and labeling of edit field and search options*. Place edit fields, option buttons and any other search element at the top of the Web page; avoid secondary elements (links, texts, banner frames, etc.). To place an object in a specific position of the visual layout, use the position CSS properties[1]. Be careful to correctly match <label for> with input elements, and place labels above or to the left of the input element, rather than below.
2. *Highlighting the search result*. Use a heading level for marking the beginning of the result list; if possible, this heading element should be the first in the page source. If a table is used to format results, a summary attribute such as "Results of the research: xxx results found" or "No results found" should be assigned. In addition, the number of the current page vs the total number of pages should be clearly indicated (e.g. x of y found).
3. *Arranging the results in numbered lists*. Put the list of the result links with their summary, just after the search result notification (nothing else should be located in the middle). Create the list by applying <ul> or <ol> elements; each item on the list must be a single result. With this feature, the screen reader informs the user of

[*] At the time of this study the author was working at the IIT-CNR.

[1] Further information on CSS language is available at the World Wide Web Consortium web pages: http://www.w3.org/Style/CSS/

the number of items; the user is then able to skip quickly from item to item. Besides, the page should not contain too many results; an appropriate number would be ten items.

4. *Recognizing sponsored links*. Sponsored links should be kept separate from the other results. Put them in a clearly labelled table (e.g. “sponsored links” summary attribute), and insert the table code after the results list in the page source; to locate sponsored links on the right side - or in another specific place – use the CSS properties.
5. *Adding navigation and help links*. Place links pointing to result pages at the end of the list (not before). This allows users to read the current results (summaries and links) first and then read the pointers to the next group of results; this is important when users move by arrow keys (i.e. in a sequential manner). Furthermore, it would be useful to add help or navigation links (in this case hidden links) for moving around the page, such as “skip to results”, “go to search edit field”, and “go to result page”.
6. *Rapid navigation*. Assign a scale of importance by the tabindex attribute so users can reach the most important elements quickly. On the first search page, higher values should be assigned to edit field and search options; whereas in the result page the higher values should be given to result links. A lower value should be assigned to secondary links if present (such as “cached” or “similar pages”). Furthermore, shortcuts may be associated with search elements (text box, buttons) and links to pages of results (e.g. accesskey=”1”, or “2”, etc.).
7. *Alerting by sound*. Different sounds for different events should provide blind users with useful information. For instance, two different sounds may be used to indicate the success (at least one result) or failure (no result) of the search.
8. *Using standards, such as aural style sheets (Cascading Style Sheets v.2).* Web designers should use aural style sheets provided by CSS2 specification for making web contents more usable and accessible to blind people. At the same time, browsers and screen readers must be able to interpret aural CSS properties.

In our study, we chose to apply these criteria to Google[2]; due to its very simple interface, it is preferred by blind people in Italy (Buzzi et al., 2004). Although the interface is already simple we aim to show how it could be further improved for easier navigation by screen reader, just by adding accessibility criteria described in the W3C Recommendations, such as structuring the content, use of CSS language, and so on. Specifically, we modified the source code of the Google home page (simple search) and the result page, while carefully maintaining the same original graphic layout, in order to not affect interaction of sighted users in any way. Henceforth, we refer to these new interfaces with the terms “Modified” Google User Interfaces (MGIs) and “Original” Google User Interfaces (OGIs). Furthermore, in (Andronico, Buzzi & Leporini, 2005) we describe the interface implementation in detail (source code), discussing how our proposed guidelines could be technically applied to Google.

Our paper is organized into five sections. After this brief introduction, in Section 2 we look at some related studies in this field. In Section 3 we will discuss in detail the interaction via the screen reader we used in our experiment, Jaws for Windows, to help an unskilled reader better understand all the functions of this assistive technology. In Section 4 we proposed two hypothetical scenarios with different blind persons performing the same task, one with the “original” and one with the “modified” Google interfaces. In this way we can better show all the steps needed during the interaction by Jaws screen reader. The final part of the paper concerns the pros and limitations of our implementation, also anticipating our new developments in this study.

2 Related work

Although many studies focus on web interface accessibility and usability (Leporini & Paternò 2004) (Nicolle & Abascal, 2001), (Scapin et al., 2000) to our knowledge few involve search engines. A great deal of research on search engines focuses on algorithms (crawling, ranking, indexing) for increasing the effectiveness and quality of results and architectures (centralized, parallel, distributed) for enhancing performance. Another branch of search engine research aims at improving the Graphical User Interface (GUI), but is unfortunately limited to visual features. An overview of possible visualizations for search query results and a discussion of the main factors for their success is included in (Mann, 1999). In (Vegas, de la Fuente & Crestani, 2003) authors present a graphical visualization tool for helping users determine the relevance of a Web page with respect to its structure. Such tools can help the sighted user decide whether a page is relevant enough to merit a visit, but unfortunately, since they are based on graphical interfaces, are useless for the sightless (who would truly benefit from this kind of support).

[2] Google, http://www.google.com/. For our study we modified the local Google interface, which is a little bit different. Actually, we regenerated the http://www.google.it interfaces for being able to carry out a user testing in Italy, but in this paper we used the http://www.google.co.uk pages for describing our work to the international community.

Search engines are particularly difficult for a blind person to use, since difficulties in Web navigation add to the complexity of the search engine's interface and functions. Specifically, for people using a screen reader (which provides a sequential access) actions take longer and tasks are more difficult since additional actions are required (Brajnik, 2004). The gap between blind and sighted users' efficiency when performing online search tasks is further explored in (Ivory, Yu & Gronemyer, 2004). In a set of experiments, the authors discovered that blind participants took twice as long as sighted users to explore search results and three times as long to explore the corresponding web pages. Goble et al. (Goble et al., 2000) introduced the model of real world travel to classify Website elements important for usability and accessibility; however they do not indicate design guidelines for developers. A specific study concerning accessibility of on-line library resources for sightless persons was performed and fully discussed in (Schmetzke, 2002).

The Web Accessibility Initiative Interest Group[3] (WAI-IG), of the World Wide Web Consortium, investigates problems in accessing web resources and produces guidelines for web content, authoring tools and user agent accessibility. A set of 14 main guidelines, called Web Content Accessibility Guidelines 1.0, has been produced by the group which is now working on version 2.0 of the Recommendation. In the United States other accessibility guidelines partially overlapping WCAG 1.0, have been defined by the Government[4] and other guidelines are specified by several national or international organizations. For an initial automatic validation several tools has been implemented[5] but in any case the most important validation requires a human control.

In 2000 the Danish National Library for the Blind, based on the WAI guidelines and their experience derived from testing Websites with assistive technology and working with blind and visually impaired users, founded the Webcenter, an on-line Center which provides tips and examples for making developers aware of the problem and stimulating the realization of accessible websites (Christensen, 2001). However, although problems have been highlighted, solutions suggested, and a validator implemented, a structured set of usability design criteria for visually impaired users has not yet been proposed.

A new approach by Donker et al. proposed an aural-based methodology for obtaining the highest possible resemblance to the visual interface. This approach introduced in (Donker et al., 2002) aims at providing access not only to the page content, but also to relevant information present in the page layout. To overcome the layout barrier, authors concentrated on the interaction possibilities in an auditory interaction realm to represent the layout of Web pages and to support the navigation by users. The auditory user interface of the system prototype was tested on seven blind users but unfortunately results were not as expected. The test revealed that the users were unable to process their tasks more effectively and efficiently with the proposed system compared to their currently used screen reader.

Another interesting research focus on search engine design for the blind is presented in (Ferworn, Bodner & Chignell, 2000). Specifically, the authors describe and discuss the implementation of an auditory search engine prototype, which provides vocal output by using real-time text-categorization to organize results into a voice menu format.

Concerning usability user testing, within the framework of the NOVA project (Non-Visual Access to the Digital Library) the Manchester Metropolitan University performed experiments on a sample of blind and visually-impaired users who carried out four information-seeking tasks, including the use of search engines. Interesting and detailed results are included in (Craven & Brophy, 2003). In (Ivory et al., 2004), a study analyzing the decision-making behavior and performance of blind and sighted users during the search task was conducted. In this study authors aim to identify page features that could be presented in results displays, and the circumstances might help users to decide whether to explore search results or not. In most cases, participants expressed a desire for additional page features, which varied depending on their visual ability and their ability to specify criteria for controlling the order of results (ranking). They also suggest various ways the user's search experience could be improved.

3 How Jaws works with a web page

In this section, for the reader's convenience, we describe how a blind user interacts with the interfaces via screen reader. The screen reader referred to in the following is Jaws for Windows (v. 5.10), which is the most often used by blind people all over the world (Damery 2003). We then used the browser IE v. 5.5 since Jaws does not work properly with other www clients, such as Netscape Navigator or Mozilla Firefox.

[3] The Recommendation developed by the WAI-IG and new works of the W3C group, are available at http://www.w3.org/TR/WCAG10/ (dated 5 May 1999) and http://www.w3.org/TR/2004/WD-WCAG20-20040730/

[4] Further information available at: http://www.section508.gov.

[5] A list of automatic validators for accessibility can be found at http://w3.org/WAI/ER/existingtools.html#Evaluation.

Jaws gives rapid access to the information which comes directly from the HTML code of the page. Practically speaking, Jaws deals with the Web page content as if it were a text document: the content is serialized as written in the html/xhtml page code, by putting each block under the other. In this context, the user can navigate within the page line by line, from top to bottom, by simply pressing the arrow keys. This is possible also thanks to a special Jaws cursor called “virtual cursor”, which allows the user to move around the page as if it were a text document. You can move the virtual cursor by Tab key, arrow keys or other special Jaws commands. When the user moves this special cursor, the screen reader announces the content encountered by the virtual cursor movements. That means there is a significant difference between that which is perceived via screen reader, due to the virtual cursor, and what is shown on the screen: how the screen reader interprets the content and how the blocks are logically sequentialized.

Also, additional information related to the document structure, such as tables, lists, headings, graphics, edit fields, buttons, and so forth, are announced to users. For example, heading tags (<h1>..<h6>), if appropriately applied, are particularly useful since they are captured and listed as an index, helping the user to navigate the page. The use of some hidden information could help users to better comprehend the page structure. To move quickly around the page content a blind user prefers to use the Tab key, for jumping from one to another interactive element (such as links, buttons, text field, etc.), rather than the arrow keys. Web developers need to take into account that due to this preference most of the text is not read, and all links content and button labels should be context-independent and self- explanatory.

A particular case occurs when we need to work with form control elements. With its virtual cursor, Jaws may handle the form elements activating a special modality – “form mode on” – by simply pressing Enter key when the virtual focus is over a form control. In this situation the screen reader informs the user with the sentence “form mode on”. If the user forgets this step, no typed text is written in a text field, no choice can be made from a combo-box, and some unwanted effect may occur (i.e. activating a search with an empty query). In fact, without “form mode on” activated and with only the Jaws virtual cursor, any key pressure performs a different command, (“t” moves to next table, “h” to next heading) and consequently we need to restart the entire desired operation.

4 Scenario Based Design: two examples

User-interaction scenarios are a descriptive story about the individuals’ experience and interaction with a particular system or application. A scenario may be a description at different level of detail, and in a finer specification, can also take into consideration hardware, software and user interface elements (Carroll, 1995). In our design process we used scenario description to understand the problems of interaction with the Google interface by screen reader and the difficulties a person may encounter during a query. This let us continue with the proposed guidelines and the restructuring of Google code.

In this study we used scenarios to illustrate the experience that two different users can have with the same task and the same Jaws screen reader, when navigating the "Original" Google Interfaces (OGIs) and the "Modified" Google Interfaces (MGIs). In particular, we intend to point out the possible benefits offered by screen readers when interacting with the new user interface. At the end of the section, we illustrate the main advantages achieved as well as a few unresolved problems.

The two scenarios described were drawn based on the personal experience of one of the authors in navigating via screen reader Jaws for Windows.

4.1 Scenario 1: interacting with OGIs

Roberto is a blind Italian boy who is not particularly skilled with computer applications. He uses the computer mainly to email his friends and navigate the Internet, visiting well-known sites. He does not know the Google search engine very well, but now he needs it to look up information about the airport in Rome. Roberto has never flown out of Rome before, and now he is going to take a trip to Amsterdam where he will attend a meeting for visually-impaired people living in Europe.

Roberto connects his computer to the Internet, opens IE, and types in the Google’s URL. As soon as the browser starts loading the Home Page, the screen reader begins to read the page content, from the very first line. Roberto stops it working because he wants to quickly reach the edit field and insert his query. He presses the Tab key and the screen reader virtual focus takes him immediately to each link focused (i.e. “images”, “groups”, “news”, “more”, as shown in lines 5, 6, 7 and 8 in the left- hand side of Figure 1) until he finds the “edit field”. Here Roberto stops, but he is unsure whether he is over the search box (the screen reader just announced “edit field”) and he decides to move up and down with the arrow keys to look for an associated label. Unfortunately there is no associated label, but only a "Google Search" button, so Robert guesses that the "edit field" encountered before was the right place to start his query, after all. Roberto activates the "form mode on" function of the screen reader (described in Section 3) and writes his query: airport Rome. Again Roberto is not sure he has written all the words correctly and checks again using a special reading command from Jaws. Finally, he presses the Enter key to start the search.

When the result page is loaded, Jaws starts again to read automatically the entire contents line by line. Roberto once again stops the process: too much information is provided and it is too hard to follow, due to the overload perceived by the user who is forced to listen to the voice synthesizer reading every single line. At this point Roberto decides to explore the page manually skipping line by line, link by link, until reaching the group of results. Pressing the Tab key repeatedly, he perceives a "new" link, which is probably a link pointing to one of the result found. Exploring again around the link with the top and down arrow keys Roberto understands that it is a sponsored link. After pressing Tab keys several times and exploring each link encountered (see the left part of Figure 2) Roberto must decide whether the result is a sponsored link or not. The whole procedure has been quite tedious and time-consuming. Finally Roberto finds the first unsponsored result, as well as another problem: there are three more links for each result ("Translate this page", "Cached" and "Similar pages") that make the exploration even more annoying: that is, one link "to be read" and three "to be skipped".

In the end Roberto finds what he was looking for (line 41 in Fig. 2), but has also encountered even more annoying difficulties.

Figure 1 reports all the steps of the Jaws interpretation during the exploration of the Google HomePage: the part on the left is produced by the original code while the right by the modified code. Parts in italics are announced by Jaws but not "explicitly" rendered on the screen (e.g. link, button, table, etc.); they are just announced to inform users about features that are perceived from a visual rendering. Parts in bold refer to the content added in the modified interface (see next section for details).

1	Google	1		New google.co.uk	
2	*Graphic* Google	2		*Graphic* Google logo	
3		3			
4	Web	4		***Heading level 2* Navigation bar:**	
5	*Link* Images	5		*Link* Navigation help **alt+h**	
6	*Link* Groups	6		Web	
7	*Link* News	7		*Link* Images	
8	*Link* More »	8		*Link* Groups	
9		9		*Link* News **alt+n**	
10	*Edit*	10		*Link* More »	
11	Google Search *Button*	11			
12	I'm Feeling Lucky *Button*	12		***Heading level 1* Searching for:**	
13		13		*Edit* **alt+s**	
14	*Link* Advanced Search	14		Google search *Button*	
15	*Link* Preferences	15		I'm Feeling Lucky *Button*	
16	*Link* Language Tools	16			
17		17		Search:	
18	Search:	18		The Web **alt+w** *Radio button checked 1 of 2*	
19	The web *Radio button checked 1 of 2*	19		pages from the UK **alt+p** *Radio button not checked 2 of 2*	
20	pages from the UK *Radio button not checked 2 of 2*	20			
21	...	21		***Heading level 2* Advanced search:**	
22		22		*Link* Advanced Search **alt+a**	
23		23		*Link* Preferences	
				Link Language Tools	
				...	

Figure 1: Google home page read by Jaws: on the left using OGIs, on the right using MGIs

4.2 Scenario 2: interacting with MGI

Sarah is a blind user who needs to make the same query as her friend Roberto, to find information about the " Rome airport". She too uses the computer and the Internet with a voice synthesized screen reader, and she is not very skilled with search tools. She now has the opportunity to use the modified Google interface to make her query.

When she opens the browser and starts loading the Google HomePage, a short sound is immediately played to alert Sarah that the screen reader will automatically read the page contents. Sarah stops this process to move manually around the page with the new shortcuts added to interact with the screen reader in the modified Google interface.

She then presses alt+s to directly move the virtual focus onto the search edit field. She activates the form modality of the screen reader to allow her to write the query, and another low sound is played. This aural feedback tells Sarah she is in the right position for inserting her query. Sarah starts writing words and for each digit inserted a little tone is heard; this confirms that she is really writing in the field. After pressing the Enter key, the result page is loaded and a short triple sound informs her that the search has succeeded: at least one result has been found. If the search fails a single deep-toned sound communicates "no result found". This feedback is particularly useful, especially when a search fails and the user must explore the page sequentially in order to discover this.

At this point Sarah can start exploring the result page, just by pressing the Tab key, which lets the focus move immediately onto the written "Results 1 - 10 of about...", at the beginning of the result section (see line 27 of Figure 2 on the right). Therefore, by simply pressing the Tab key once, Sarah is located within the result area and knows the number of results. At this point, pressing the Tab key, she visits the first result link (line 31). She notes that it is an Italian site and decides to jump to the next result. By another Tab key the focus reaches the link "ADR - Aeroporti di Roma Official Site – Homepage", i.e. the second result (line 41) since the links "translate this page", "Cached" and "Similar pages" are skipped. If Sarah decides to explore around the achieved link, she notes that the links are numbered: this is useful for better orientation among results. Furthermore, she can also explore the following results pages (Roberto did not even consider this, since he would have needed to press several tab keys) simply using alt+plus key. Anyway, Sarah can skip directly to the "result page" section by simply pressing "h" key (a Jaws specific command), that lets her choose the section name from the heading list generated by Insert+f6 (another Jaws specific command).

Sarah ends her search successfully, without any frustration. She needed to learn a few additional commands for the new interface, but the new page structure makes her experience with the application easier than the one in the Roberto scenario.

Figure 2 reports all steps of the Jaws interpretation during the exploration of the Google result page: the part on the left is with the original code while the right is with the modified code.

1	Google Search: airport rome	1	Result page for airport rome
2		2	
3	*Link Graphic* Go to Google Home	3	*Link Graphic* Go to Google Home **alt+g**
4	Web	4	**Heading level 1 Navigation bar:**
5	*Link* Images	5	***Link* Navigation Help alt+h**
6	*Link* Groups	6	Web
7	*Link* News	7	*Link* Images
8	*Link* More »	8	*Link* Groups
9		9	*Link* News **alt+n**
10	*Edit* airport rome	10	*Link* More »
11	Google Search *Button*	11	
12	I'm Feeling Lucky *Button*	12	**Heading level 2 Searching for:**
13		13	*Edit* airport rome **alt+s**
14	*Link* Advanced Search	14	Google Search *Button*
15	*Link* Preferences	15	I'm Feeling Lucky *Button*
16		16	
17	Search:	17	Search:
18	The web *Radio button checked 1 of 2*	18	The Web **alt+w** *Radio button checked 1 of 2*
19	pages from the UK *Radio button not checked 2 of 2*	19	pages from the UK **alt+p** *Radio button not checked 2 of 2*
20		20	
21	Web	21	***Heading level 2* Advanced search:**
22	Results 1 - 10 of about 1,080,000 for airport rome. (0.12	22	Advanced Search **alt+a**
23	seconds)	23	*Link* Preferences
24		24	
25	*Table with 4 columns and 6 rows*	25	Web
26	Sponsored Links	26	**Heading level 1 Search results:**
27	*Link* Rome Airport	27	Results 1 - 10 of about 1,080,000 for airport rome. (0.12
28	[... other 7 links and extracted texts of sponsored links...]	28	seconds)
29	*table end*	29	
30		30	**1**
31	*Link* ADR - Sito ufficiale Aeroporti di Roma - Homepage -	31	*Link* ADR - Sito ufficiale Aeroporti di Roma - Homepage -
32	[*Link* Translate this page]	32	[*Link* Translate this page]
33	Sito ufficiale della Società Aeroporti di Roma.	33	Sito ufficiale della Società Aeroporti di Roma. Informazioni,
34	Informazioni, notizie e rubriche su azienda, aeroporti di	34	notizie e rubriche su azienda, aeroporti di Fiumicino e Roma

35	Fiumicino e Roma Ciampino, orario voli, compagnie aeree	35	Ciampino, orario voli, compagnie aeree ...
36	...	36	http://www.adr.it/ - 43k -
37	http://www.adr.it/ - 43k -	37	*Link* Cached -
38	*Link* Cached -	38	*Link* Similar Pages
39	*Link* Similar pages	39	
40		40	**2**
41	*Link* ADR - Aeroporti di Roma Official Site - Homepage-	41	*Link* ADR - Aeroporti di Roma Official Site - Homepage -
42	[*Link* Translate this page]	42	[*Link* Translate this page]
43	... Airports Eight new commercial activities have been	43	... Airports Eight new commercial activities have been
44	recently opened at the two Roman airports with the aim to	44	recently opened at the two Roman airports with the aim to
45	diversify the commercial offer to the passengers.	45	diversify the commercial offer to the passengers. ...
46	...www.adr.it/default.asp?L=3 - 41k	46	http://www.adr.it/default.asp?L=3 - 41k -
47	*Link* Cached -	47	*Link* Cached -
48	*Link* Similar pages	48	*Link* Similar pages
49		49	
50	*Link* Rome Fiumicino Airport	50	**3**
51	[*Link* Translate this page]	51	*Link* Rome Fiumicino Airport -
52	...HubPage: Airport Guide: Rome Fiumicino Airport: Rome	52	[*Link* Translate this page]
53	Fiumicino Airport. Airwise - The Airport and Air Travel	53	...HubPage: Airport Guide: Rome Fiumicino Airport: Rome
54	Guide. ... Rome Fiumicino Airport News. Rome Hotels. ...	54	Fiumicino Airport. Airwise - The Airport and Air Travel
55	http://www.airwise.com/airports/europe/FCO/FCO_01.html	55	Guide. ... Rome Fiumicino Airport News. Rome Hotels. ...
56	- 14k	56	http://www.airwise.com/airports/europe/FCO/FCO_01.html
57	*Link* Cached -	57	- 14k
58	*Link* Similar pages	58	*Link* Cached -
59	[... other result links...]	59	*Link* Similar pages
60		60	[...other result links...]
61	Result Page:	61	**Heading level 3** Sponsored links
62	1	62	[... Links and extracted text of sponsored links...]
63	*Link* 2	63	**Heading level 1** Result page:
64	*Link* 3	64	1
65	...	65	*Link* 2
66	*Link* Next	66	*Link* 3
67	…	67	...
68		68	*Link* Next **alt++** …

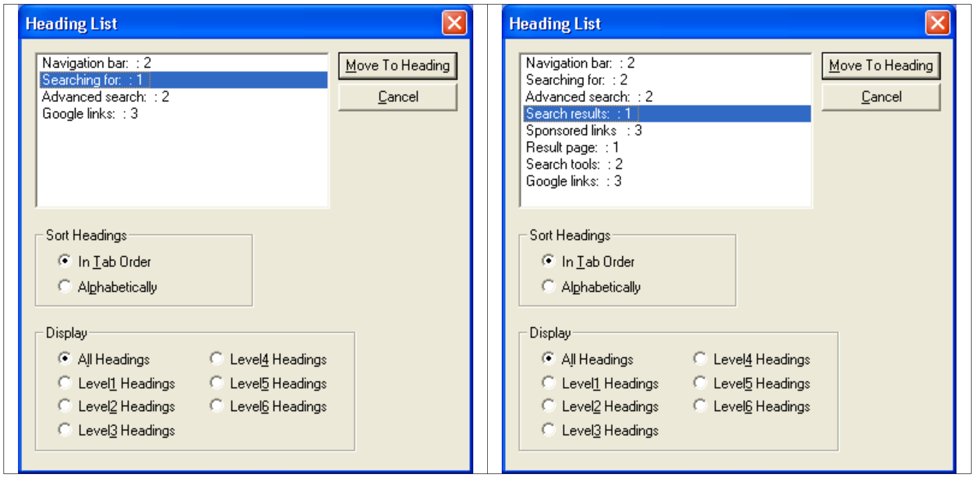
Figure 2: Google result page read by Jaws: the original page on the left, the modified one on the right

4.3 Further discussion of the two scenarios

In the previous section we showed how the Google interface could affect the experience of a blind user during navigation. Designing the Modified Google Interfaces, we properly structured the interface and added some useful mechanisms for improving the exploration of a page through keyboard commands with the screen reader. We have:

- arranged the page content in a logical order by <div> blocks rather than tables;
- structured the page content by heading levels with some hidden labels "capturable" by the screen reader;
- added index levels and shortcuts to give a different visiting order to the most important elements;
- added sounds for notifying the user that a certain event is happening.

As we see in the two scenarios, Roberto's and Sarah's experiences differ greatly. Facilitating page navigation also reduces a blind user's sense of frustration. Since nothing has changed in the visual interface, a sighted person perceives nothing unusual. Instead a good code can offer an easier way for blind users to navigate around a web page. In particular, adding heading levels with hidden labels permits the user a quick overview of the page structure and the relative sections (see Figure 3).



Navigation bar: : 2
Searching for: : 1
Advanced search: : 2
Google links: : 3

Sort Headings
● In Tab Order
○ Alphabetically

Display
● All Headings
○ Level1 Headings
○ Level2 Headings
○ Level3 Headings
○ Level4 Headings
○ Level5 Headings
○ Level6 Headings

Navigation bar: : 2
Searching for: : 2
Advanced search: : 2
Search results: : 1
Sponsored links : 3
Result page: : 1
Search tools: : 2
Google links: : 3

Sort Headings
● In Tab Order
○ Alphabetically

Display
● All Headings
○ Level1 Headings
○ Level2 Headings
○ Level3 Headings
○ Level4 Headings
○ Level5 Headings
○ Level6 Headings

Figure 3: Logical sections of MGIs: a) home page and b) results page (generated by a specific Jaws command)

In our proposal, thanks to the new structure blind users can jump to a specific part of the interface using three different Jaws commands:

- By pressing the Insert+f6 key, the user obtains the whole list of the headings used. With the arrow keys s/he can select the desired section just by pushing the Enter key.
- The "h" and "shift+h" commands of Jaws permit users to move to the next or previous heading in the list (whatever the level is), allowing the user explore the page sections sequentially.
- By pressing the "1", "2" … "6" keys, users jump directly to the next Heading at Level (or to the previous heading if *shift key* is also associated). For instance, if the user presses the "1" key when the results page is loaded, the current focus moves to the "search result section", since a tag <h1> has been associated to this section (as highlighted in Figure 3b). In any case, other specific commands for skipping to the first or last heading are available.

Relating to different position of blocks, note the difference in reading the page content sequentially (Fig. 1 and Fig. 2). On the home page of the original interface, when the user reads search fields and would like to find search options, they could become confused because after the search buttons there is the link for "advanced search"; the radio buttons to set where to carry out the search are below and users may be not able to find them. This does not occur in the modified interface where the radio buttons come before the "advanced search" link (see Figure 1). In the same way, "Sponsored links" are simply placed in a lower position. You can easily reach them if you want to explore that particular group of results.

4.4 Open Issue

Although it simplifies interaction for blind persons, our proposed interface still has some limitations. First, respecting the original Google interface, the search fields and options blocks are repeated in the results page. A user interacting via screen reader and especially by voice synthesizer would like to read only the results, so the search fields and options should be placed only at the bottom of the page. However, in our interface, we allow the user to skip directly to the "search result" section. If the user moves sequentially around the result page, they read the same information as in the original page (plus few added information); but if he/she moves by Tab key or special screen reader commands, it is more efficient.

Another issue concerns the link content and extracted text of the found results, but this problem is sharted with sighted users as well. The search engine should be structured so that the result links and extracted text have significant content.

Concerning implementation, due to Jaws screen reader limitations, our interface runs well with IE but not with Netscape and Mozilla. We used Javascript to activate different sounds for communicating important events immediately to the user, such as "the focus is on the search box" or whether the search query produced results or not. Obviously this additional feature is not accessible if the user uses a textual browser such as Lynx or a very old browser or if Java script is disabled. When browsers and screen readers are able to interpret aural CSS properties this type of feature would be available with the appropriate style sheet.

5 Conclusions and future works

In our research we modified Google User Interfaces to make this tool easier to use. The constraint for our implementation was to maintain the same graphical layout in order to make the source modification transparent for sighted user interaction. Our approach considers the difference between visual layout and aural perception via screen reader, trying to convey the same information provided by visual elements (such as position, color, white spaces, etc.) by using standard language elements. Specifically we use XHTML source code and CSS for separating structure from content, java script for adding sounds which notify the user that a certain event is happening, hidden labels to orient the user in the logical section of the interface (navigation bar, search box and buttons, results, sponsored results and so on), heading levels for highlighting the most important parts and permit easy moving to this point, and lastly, Tab keys and access keys for simplifying navigation around the interface. In addition we also move the most interesting elements (from a user's point of view) to the beginning of the source code arranging the correct visual position on the interface with the CSS property (i.e. position) and eliminated tables used for layout (using DIV elements instead).

In this paper we have discussed two scenarios showing interactions via screen reader for the original and the modified Google Interfaces. Comparing the interaction via screen reader, we showed that interfaces following our design principles enable blind users to navigate the interface more quickly and efficiently, furnishing better orientation within the page content. Our application demonstrated that with relatively little effort it is possible to increase the accessibility of interfaces while keeping an appealing graphical layout.

As the next stage of our research, we intend to perform a test with blind users to verify the effect of our Modified Google Interfaces on interaction via screen reader.

References

Andronico P., Buzzi M., Leporini B. (2005). Improving Search Engine Interfaces for Blind Users: a Case Study, To appear in International Journal Universal Access in the Information Society.

Buzzi M., Andronico P., Leporini B. (2004). Accessibility and Usability of Search Engine Interfaces: Preliminary Testing. In Adjunct Proceedings of 8th ERCIM UI4ALL Workshop, Vienna, Austria, from http://www.ui4all.gr/workshop2004/publications/adjunct-proceedings.html

Brajnik G. (2004) Achieving Universal Web Access through Specialized User Interfaces. In Adjunct Proceedings of 8th ERCIM UI4ALL Workshop. Vienna, Austria, from http://www.ui4all.gr/workshop2004/publications/adjunct-proceedings.html

Carroll John M. (1995) Scenario-Based Design, John Wiley and Son, USA.

Craven, J., Brophy, P. (2003). Non-visual access to the digital library: the use of digital library interfaces by blind and visually impaired people. Technical report, Manchester: Centre for Research in Library and Information Management, from http://www.cerlim.ac.uk/pubs/index.php

Christensen S., (2001). How we work to make the Web speak. *Computers in libraries*, 21 (9), 30-34, from http://www.infotoday.com/cilmag/oct01/christensen.htm

Damery E. (2003). JAWS for Windows version 4.51. New features and accessing the Internet. In Proceedings of 18th Technology and Persons with Disabilities Conference (CSUN), Los Angeles, California, USA.

Donker H., Klante P., Gorny P. (2002). The Design of Auditory User Interfaces for Blind Users. In Proceedings of the Second NordiCHI, Tradition and Transcendence, Edited by O. W. Bertelsen, S. Boedker and K. Kuutti, October 19-23, 2002, Aarhus Denmark. New York 2002: ACM Press, pp. 149-155.

Ferworn, A., Bodner, R., Chignell, M. H. (2000). Auditory WWW Search Tools. In Proceedings. of International Conference on Auditory Display. Atlanta, USA, from http://www.icad.org/websiteV2.0/Conferences/ICAD2000/ICAD2000.html

Goble C., Harper S., Stevens R. (2000). The Travails of Visually impaired Web Travellers, In Proceedings of ACM Hypertext 2000, ACM Press, pp. 1-10.

Ivory, M. Y., Yu, S., Gronemyer, K. (2004). Search result exploration: a preliminary study of blind and sighted users' decision making and performance. In Extended abstracts of the Conference on Human Factors in Computing Systems (CHI 2004), pp. 1453-1456.

JAWS for Windows, Freedom Scientific, from http://www.freedomscientific.com

Leporini, B., Paternò, F. Increasing (2004). Usability when Interacting through Screen Readers, International Journal Universal Access in the Information Society, special Issue on "Guidelines, standards, methods and processes for software accessibility", Springer Verlag, Vol.3, N.1, pp. 57-70.

Leporini, B., Andronico P., Buzzi. (2004). Designing Search Engine User Interfaces for the visually impaired. ACM International Cross-Disciplinary Workshop on Web Accessibility 2004, NY, USA, pp 57-66.

Mann, T.M. (1999). Visualization of WWW-Search Results. In In Proceedings of International Workshop on Database & Expert Systems Applications. pp. 264-268.

Nicolle, C., Abascal J. (2001). Inclusive design guidelines for HCI, p. 285, Taylor & Francis.

Scapin, D., Leulier, C., Vanderdonckt, J., Mariage, C., Bastien, C., Farenc, C., Palanque, P., and Bastide, R. (2000). In Proceedings of Conference on Human Factors and the Web. Austin, Texas. http://www.tri.sbc.com/hfweb/scapin/Scapin.html.

Schmetzke, A. (2002). The Accessibility of Online Library Resources for People with Print Disabilities: Research and Strategies for Change. In Proceedings of International Conference Computers Helping People with Special Needs, Linz, Austria, pp. 390-397.

Vegas, J., de la Fuente, P., Crestani, F. (2003). WebDocBall: A Graphical Visualization Tool for Web Search Results. European Conference on Information Retrieval. Springer LNCS 2633, pp. 351-362.