Adopting a Scenario-Based Approach to Test a Modified Search Engine User Interface 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

Search engines have become an indispensable tool for retrieving information from the Internet; thus simple and effective interaction is important for anyone and especially for blind users who navigate via screen reader. We believe that an appropriately designed user interface is crucial for improving accessibility and usability. For this reason we proposed a set of specific guidelines that would improve interaction of the blind with search engines and applied them to Google, modifying the source code of its simple search interfaces while maintaining the same graphical layout.
In this paper we discuss user interaction with both the original and modified Google UIs, presenting two scenarios of use in which two blind persons were assigned the same search task. In both scenarios the modified UIs improve the sequential access furnished by screen reader since the content's internal structure has changed by moving at the top of the page source file the parts that the user judges more important, i.e. the search box on the home page and the query results in the result page.

1 Introduction

Since 2004 we have been working on a research project aimed at improving search engine usability for the blind. This paper is an extension of [2], which defines a scenario of use of the modified and the original Google UIs. This extension includes: the user's interaction in case of unsuccessful search, and an update of the scenario of use due to an additional source code optimization. In fact, in [1], we described the interface implementation in detail (source code), discussing how our proposed guidelines could be technically applied to Google. After this we tested the prototype with a blind user that led us to another important optimization to be applied to the modified UIs. The iterative design implementation cycle brought us to move the query results to the top of the result page body content. For this reason we added a new empirical guideline (*Place the most important elements of the interface at the top of the source file*), very important for anyone using basic screen reader commands and sequentially moving via Tab key.

Search engine design guidelines

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 the interface and execute a task. Our focus is on aural perception vs visual experience. The screen reader in fact deals with Web page content in a manner that differs greatly from visual rendering. Interaction requires a certain expertise in advanced screen reader and browser commands, and orienting oneself within the page contents can be frustrating. 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.
Considering the many difficulties that accompany screen reader interaction, we have proposed [15] the following eight guidelines for improved search engine UIs, in order to simplify navigation and perception of the whole page content:

1. *Easy location and labeling of edit field and search options.*
2. *Highlighting the search result.*
3. *Arranging the results in numbered lists.*
4. *Recognizing sponsored links*.

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

5. *Adding navigation and help links.*
6. *Rapid navigation.*
7. *Alerting by sound.*
8. *Using standards, such as aural style sheets (Cascading Style Sheets v.2).*

As previously mentioned, a new guideline was added:

9. *Place the most important elements of the interface at the top of the source file*. For search engines the most important elements are: search field and options, and query results.

We chose to apply these criteria to Google since its interface, although already simple, would be further improved for easier navigation by screen reader, just by adding accessibility guidelines described in the W3C Recommendations, such as structuring the content, use of CSS language, and applying usability criteria.
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 the 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).
Our paper is divided into five sections. After this brief introduction, Section 2 presents some related studies in this field. In Section 3 we describe the main redesign steps of the "modified" Google interfaces. In Section 4 we hypothesize two scenarios with different blind persons performing the same search task, one with the "original" and one with the "modified" Google interfaces, and show all steps needed for interaction via screen reader. Last, the conclusion analyses advantages and limits of our implementation.

2 Related work

Although many studies focus on web interface accessibility and usability, including studies regarding people with special needs ([14], [18], [19]) to our knowledge few works have addressed 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 [16]. In [21] the 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 blind. 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 [4]. The gap between blind and sighted users' efficiency when performing online search tasks is further explored in [12]. 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. [11] introduced the model of real world travel to classify the importance of Website elements for reaching usability and accessibility; however they do not indicate design guidelines for developers.
The Web Accessibility Initiative Interest Group[1] (WAI-IG), of the World Wide Web Consortium, investigates problems 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, which takes more into greater account usability. In the United States other accessibility guidelines, partially overlapping WCAG 1.0, have been defined by the Government[2] and other guidelines are specified by several national or international organizations. Several tools have been implemented[3] for an initial automatic validation but in any case the most reliable validation still requires a human control.
In 2000 the Danish National Library for the Blind, based on the WAI guidelines and their experience 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

[1] 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/
[2] Further information available at: http://www.section508.gov.
[3] A list of automatic validators for accessibility can be found at http://w3.org/WAI/ER/existingtools.html#Evaluation.

creation of accessible websites [7]. However, although problems have been highlighted, solutions suggested, and a relevant information present in the page layout. To overcome the layout barrier, authors concentrated on the interaction validator implemented, a structured set of usability design criteria for visually impaired users has not yet been proposed. Further, a specific study concerning accessibility of on-line library resources for the blind was performed and fully discussed in [20].
A new approach by Donker et al. proposed an aural methodology for obtaining the greatest resemblance to the visual interface. This approach introduced in [9] aims at providing access not only to the page content, but also to 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 study is presented in [10], where 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 performed four information-seeking tasks, including the use of search engines. Interesting and detailed results are included in [6]. In [12], a study analyzed the decision-making behavior and performance of blind and sighted users during the search task. In this study authors aimed at identifying 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 Modifying Google UIs

We chose to modify http://www.google.co.uk/ and http://www.google.co.it/, i.e. the regional version of Google for two reasons: regional versions are used locally (google.com in fact redirects the browser to its regional version) and their UIs are a little bit more complicated with respect to the .com version. The home page provides additional options (radio buttons) to restrict the search to the specific country domain (.uk, .it, etc.) or language (if different from English), and the result page, for not English-speaking countries, presents, for each result, an additional link [translate this page], so user interaction is different.
Designing the Modified Google Interfaces, we properly structured the interface and added some useful mechanisms for improving page exploration through keyboard commands with the screen reader. Specifically we:

- arranged the page content in a logical order by 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. Specifically sounds were associated to relevant events such as: focus on the search edit field and on the search outcome (results found or not).
- accessory elements such as hidden labels and result numbering were added as well, to further simplify user interaction and user orientation.

3.1 Logical sections

As regards the code, we used XHTML and CSS properties for separating content from its rendering; we replaced tables by using <DIV> elements instead, and the CSS position property for arranging the object in specific areas of the window for visual rendering.
Specifically restructuring the code implied defining logical sections of the interface by grouping sets of homogenous text and elements together and by giving the user the possibility of jumping easily from part to another, structuring these logical sections by heading levels.
Furthermore, in the environmental testing phase, another optimization was carried out: the most important interaction elements for a search engine – i.e. the "search box" in the home page and the "results" in the result page - have been pushed up to the first position into the source code, by using an absolute positioning through a CSS property used with tags. This means giving a different order to page sections compared to the way they are visualized on the screen. Indeed, initially [1] we decided not to use an absolute positioning (i.e. the CSS property position: absolute) due to browsers unable to correctly handling this property. Next, we decided to use that property

in order to test if there are some effective advantages in exploring a Web page where blocks are arranged sequentially according to a certain importance level. The goal is facilitating identification of the most important interaction elements of a search engine by a screen reader (which reads the page content sequentially). In fact, a study conducted by Nielsen [17] revealed that users have specific expectations for the behavior of search. Designs that invoke this mental model but work differently are confusing. Users expect search to have three components: (1) A box where they can type words; (2) A button labeled "search" that they click to run the search; (3) A list of top results that is linear, prioritized, and appears on a new page.
Figure 1 shows a code fragment, which reports the order given to the sections available in the result page. In order to locate all the blocks so that they appear correctly – i.e. so that the visual layout remains the same – the CSS property “position: absolute” has been used. In practice, all blocks with an absolute positioning have been located within a container block with a relative positioning in order to reduce the compatibility problems with the browsers regarding the CSS property.

```
<!-- Result page -->
<div id="big-section">

<!-- Result section -->
<div id="result-section">
...
<h1 class="hidden-label"> Search results: </h1>

<div id="results">
        <!-- results block contents all the results and sponsored links -->
</div> <!-- end of the result block -->

<div id="result-pages">...</div> <!-- end of pages block -->
</div> <!-- end of result-section -->

<div id="search-section">
<!-- Search Google elements: text field, radio buttons, search button -->
</div> <!-- end of search-section -->

<div id="navigationbar">
        <h2 class="hidden-label"> Navigation bar </h2>
        <!-- this part is visualized on the top right -->
</div> <!-- end of navigation bar -->

<div id="bottom"> <!-- Bottom -->
<!-- In this sections bottom page elements are visualized: Advanced search, Preferences, copyright, etc.-->
</div> <!—end of  bottom -->
</div> <!-- end of big-section -->
```

Figure 1 - XHTML code portion showing a correct block sequence assigned to blocks

The simple search in the MGI was structured in four sections, repositioned in the source file as follows (see **Figure 2**, right side).

1. Search box and options
2. Advanced search and preference links
3. Navigation bar
4. Google info and other links.

The page of query results was structured in seven sections, sequentialized in the source file as follows (see Figure 5, right side):

1. Results
2. Sponsored links
3. Result Pages (previous, next, numbered pages)
4. Search box and options

5. Advanced search and preference links
6. Navigation bar
7. Google links (Google home, advertising programs, etc.).

Thanks to sections' re-positioning as well as their association with heading levels, users can get on-fly a kind of "page index" by pressing the Jaws command Insert+f6 (available only in Internet Explorer and Jaws 4.5 or higher). In this way users glance at the page content as well as skip quickly from one section to another as described in Section 4.4 (see Figure 6).

3.2 Tab and Access Keys

Important interface elements were made available via shortcuts. Access keys were associated with relevant objects, such as search edit box, advanced search, next result page and previous result page links. A navigation help page – reachable by a hidden-link with the shortcut alt+h - was added for supporting beginning users on interface features.

Table 1 - Shortcuts in the modified Google UIs

Key	**Description**
H	Navigation help
G	Google home page
0	Search field
W	Radio button "the Web"
P	Radio button «Pages from UK»
A	Advanced search
L	Language tools
N	News
+	Next page
-	Previous page

Further, since moving by Tab key is often used by the blind for positioning more quickly on a relevant part of the page, we also defined an empirical visiting order of links via Tab key. A finer granularity was applied, according to the following sequence:

1. Search results status (i.e. Results 1 - 10 of about... or no results);
2. First result, Second result, etc. (at this level cached and similar links are skipped. User can access this secondary links of the result explored with the arrow keys);
3. Result pages (Prev, 1, 2,…, Next);
4. Search Tools (i.e. "Search within results" and "Search Tips");
5. Sponsored Links;
6. Searching for (simple search box and options);
7. Advanced Search and preferences;
8. Navigation bar;
9. Google Info and other links;
10. Cached and similar page links.

Practically speaking, an "importance" order is given to the elements (i.e. links) in order to "drive" the user's navigation.

It is clear that our choices aimed to optimize user interaction when navigating via keyboard. Search engine companies would decide on the order to visit elements and logical sections, according to their needs (for instance sponsored links may be announced by screen reader before those of results). Our point of view took into account users' preferences, explored in [1].

For details concerning the source code of the modified interfaces and for examples, the reader may refer to [1].

3.3 Auditory support

To improve a blind person's interaction with the user interface, short sounds have been added to the interface. In particular we added four types of sounds:

- A double short musical sound composed of two ascending piano notes is played when the focus is over the search edit field. This sound is played every time the Google home page is loaded as the focus moves directly to the search edit field; besides, the sound is played every time user moves through the Tab key and the focus is over the edit field.

- A triple ascending sound is played when the search process is successful, i.e. at least one result is returned.
- A single deep (low) sound to inform the user of a failed search process (no results).
- A very short sound (just a beep) has been added to the two radio buttons “the Web” and “pages from the UK”. Each radio button has a short sound with a different tonality for better identifying them.

Thanks to these sounds, users can perceive more quickly certain events that occur.

4 Scenario Based Design: two examples

A user-interaction scenarios is a descriptive story about the individuals’ experience and interaction with a particular system or application. A scenario may be a description at a different level of detail, and in a finer specification, can also take into consideration hardware, software and user interface elements [5]. In our design process we used a scenario description to understand the problems of interaction with the Google interface by screen reader and the specific difficulties a person may encounter during a query. This let us continue with the proposed guidelines and the restructuring of the Google code.
In this study we used scenarios to illustrate the experience of two different users 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 improvments as well as a few unresolved problems.
The two scenarios were based on the personal experience of one of the authors in navigating via the screen reader Jaws for Windows.

4.1 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 worldwide [8]. 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.
Jaws gives rapid access to the information, which comes directly from the page code. 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 around 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 what 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 relating to the document structure, such as tables, lists, headings, graphics, edit fields, buttons, and so forth, are announced to users. For example, if appropriately applied, heading tags (<h1>..<h6>) 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 the Enter key when the virtual focus is over a form element. 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 operation.

4.2 Scenario 1: interacting with OGIs

Roberto is a blind 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 an european meeting for visually-impaired people.

Roberto connects his computer to the Internet, opens IE, and types in the Google 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 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”, “froogle”, “more”, as shown in lines 5, 6, 7, 8 and 9 in the left-hand side of **Figure 2**) 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 4.1) 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 (Insert+Up Arrow). Finally, he presses the Enter key to start the search.

When the result page is loaded, Jaws starts again to automatically read the entire contents line by line. Roberto once again halts 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.

After pressing the Tab key many times (Figure 3, left side), Roberto understands he made an error: he typed the string “air portrome” instead of “airport rome” thus the query had produced no results.

Roberto corrects the keywords and starts again the query. Once again he stops the screen reading and pressing the Tab key repeatedly, he perceives a “new” link, which is probably a link pointing to one of the results found. Again exploring around the link with the up 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 4) Roberto must decide whether the result is a sponsored link or not. The whole procedure has been tedious and time-consuming. Finally Roberto encounters the first non-sponsored 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”. This link is in Italian language so Roberto continue his exploration.

In the end Roberto finds what he was seeking (line 51 in Figure 5), but has also encountered even more annoying difficulties.

Figure 2 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.

1	Google	1	Modified google.uk	
2	*Graphic* Google	2	*Graphic* Google Logo	
3		3		
4	Web	4	**Heading level 1 Search:**	
5	*Link* Images	5	*Edit* **alt+c**	
6	*Link* Groups	6	Google Search *Button*	
7	*Link* News	7	I'm Feeling Lucky *Button*	
8	*Link* Froogle	8		
9	*Link* more »	9	Search:	
10		10	*Radio button checked* the Web **alt+w**	
11	*Edit*	11	*Radio button not checked* pages from the UK **alt+p**	
12		12		
13	Google Search *Button*	13	**Heading level 2 Advanced Search:**	
14	I'm Feeling Lucky *Button*	14	*Link* Advanced Search **alt+a**	
15		15	*Link* Preferences	
16	*Link* Advanced Search	16	*Link* Language Tools **alt+l**	

17	*Link* Preferences	17		
18	*Link* Language Tools	18	**Heading level 2 navigation bar:**	
19		19	*Link* Navigation help **alt+h**	
20	Search:	20	Web	
21	*Radio button checked* the web	21	*Link* Images	
22	*Radio button not checked* pages from the UK	22	*Link* Groups	
23		23	*Link* News **alt+n**	
24	*Link* Advertising Programs -	24		
25	*Link* Business Solutions -	25	**Heading level 2 Google links:**	
26	*Link* About Google -	26	*Link* Advertising Programs -	
27	*Link* Go to Google Com	27	*Link* Business Solutions -	
28		28	*Link* About Google -	
29	©2005 Google	29	*Link* Go to Google Com	
30		30		
31		31	Modified Google Interface	

Figure 2: Google home page read by Jaws: on the left using OGIs, on the right using MGIs

4.3 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 the Tab key to move towards the search field. By pressing just one time the Tab key, the virtual focus moves directly onto the search edit field. It is enough to press the Tab key once, because the Edit field is the first interactive element encountered by the focus when navigating within the page. Moreover, the screen reader announces the sentence "Searching for", which is the label assigned to the edit field. At this point, Sara activates the form modality of the screen reader to allow her to write the 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. Sara types "air portrome" and presses the Enter key.

Soon, a hollow sound is heard. Sara understands some mistakes were made in writing the search query. When she starts to explore Web page by arrow keys, the search failure is immediately encountered: Jaws reads "No results found" at line 5 (See Figure 3 on the right). Note that Roberto in the same situation had to read many lines as the "no result" message was localized at 22st line (See Figure 3 on the left). Proceeding to navigate the page content, Sara locates the wrong search text and recognizes her error: in the search text "air portrome" the space between the two words is not at the right place. So, Sara presses the Tab key and the focus moves onto the search edit field. She activates the "form modality" (see Section 4.1) in order to rewrite the correct search string. Sara types the "airport rome" text and starts searching. After pressing the Enter key, the result page is loaded and a short triple ascending sound informs her that the search has succeeded: at least one result has been found. This sound differs from the one used when the search fails, i.e. a single deep-toned sound communicates "no result found". Noting that this aural feedback is particularly useful, especially when a search fails and the user must explore the page sequentially in order to discover this.

1	Air portrome - Google Search	1	No results - Modified google.uk	
2		2		
3	*Link* Google homepage	3	Web	
4		4		
5	Web	5	**Heading level 1 No results:**	
6	*Link* Images	6	Your search - - air portrome - did not match any	
7	*Link* Groups	7	documents.	
8	*Link* News	8	Suggestions:	
9	*Link* Froogle	9	Block quote start	
10	*Link* more »	10	- Make sure all words are spelled correctly.	
11		11	- Try different keywords.	
12	*Edit* air portrome	12	- Try more general keywords.	
13	Search *Button*	13	Block quote end	
14		14		

15	*Link* Advanced search	15	**Heading level 2 Search:**	
16	*Link* Preferences	16	*Edit* air portrome **alt+c**	
17		17	Search *Button*	
18	Search:	18		
19	*Radio button checked* the Web	19	Search:	
20	*Radio button not checked* pages from the UK	20	*Radio button checked* the Web alt+w	
21		21	*Radio button not checked* pages from the UK alt+p	
22	Web	22		
23		23	**Heading level 2 Advanced Search:**	
24	Your search – air portrome - did not match any	24	*Link* Advanced Search alt+a	
25	documents.	25	*Link* Preferences	
26		26		
27	Suggestions:	27	**Heading level 2 Navigation bar:**	
28	Block quote start	28	*Link* Google Homepage ***alt+g***	
29	- Make sure all words are spelled correctly.	29	Web	
30	- Try different keywords.	30	*Link* Images	
31	- Try more general keywords.	31	*Link* Groups	
32	Block quote end	32	*Link* News **alt+n**	
33		33	*Link* Froogle	
34	copy2005 Google	34	*Link* more	
35		35		
36		36		
37		37	Modified Google Interface	

Figure 3 – "No result" page read by Jaws: the original version on the left, the modified one on the right

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 6 of Figure 5 on the right). Therefore, by simply pressing the Tab key once, Sarah is located within the result area and learns the number of results. At this point, pressing the Tab key, she visits the first result link (line 10). 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 20) 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, with no 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.

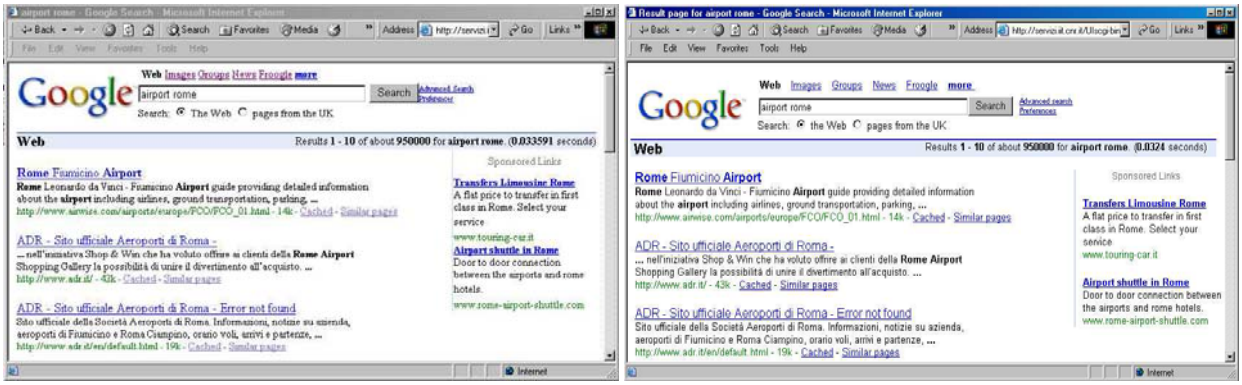

Figure 4: **Google result page: the original page on the left, the modified one on the right**

Figure 5 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	airport rome - Google Search	1	Result page for airport rome - Modified google.uk
2		2	
3	*Link* Vai alla pagina principale di Google	3	Web
4		4	
5	Web	5	**Heading level 1 Results:**
6	*Link* Images	6	Results 1 - 10 of about 950000 for airport rome. (0.0324 seconds)
7	*Link* Group	7	
8	*Link* News	8	
9	*Link* Froogle	9	**1**
10	*Link* More	10	*Link* ADR - Sito ufficiale Aeroporti di Roma
11		11	
12	*Edit* airport rome	12	Aeroporto Intercontinentale Leonardo da Vinci di
13	Search Button	13	Fiumicino e Giovan Battista Pastine di Ciampino. Le
14		14	infrastrutture, la mappa, i voli e le tariffe.
15	*Link* Advanced Search	15	http://www.adr.it/ - 43k -
16	*Link* Preferences	16	*Link* Cached -
17		17	*Link* Similar pages
18	Search:	18	
19	*Radio button checked* the Web	19	**2**
20	*Radio button not checked* pages from the UK	20	*Link* ADR - Aeroporti di Roma Official Site -
21		21	
22	Web	22	The company controlling Rome's two airports -
23		23	Fiumicino and Rome Ciampino - provides airport
24	Results 1 - 10 of about 950000 for airport rome.	24	maps and information on flights and facilities, ...
25	(0.033591 seconds)	25	http://www.adr.it/default.asp?L=3 - 42k -
26		26	*Link* Cached -
27	*Table with 4 columns and 5 rows*	27	*Link* Similar pages
28		28	
29	Sponsored links	29	**3**
30		30	*Link* ADR - Sito ufficiale Aeroporti di Roma - Error
31	*Link* Transfers Limousine Rome	31	not found
32	A flat price to transfer in first class in Rome. Select	32	
33	your service	33	Sito ufficiale della Società Aeroporti di Roma.
34	www.touring-car.it	34	Informazioni, notizie su azienda, aeroporti di
35		35	Fiumicino e Roma Ciampino, orario voli, arrivi e
36	*Link* Airport shuttle in Rome	36	partenze, ...
37	Door to door connection between the airports and	37	http://www.adr.it/ - 19k -
38	rome hotels.	38	*Link* Cached -
39	www.rome-airport-shuttle.com	39	*Link* Similar pages
40		40	
41	*table end*	41	... [other results]
42		42	
43	*Link* ADR - Sito ufficiale Aeroporti di Roma	43	***Heading level 3*** Sponsored Links
44	Aeroporto Intercontinentale Leonardo da Vinci di	44	
45	Fiumicino e Giovan Battista Pastine di Ciampino.	45	*Link* Transfers Limousine Rome
46	Le infrastrutture, la mappa, i voli e le tariffe.	46	A flat price to transfer in first class in Rome. Select
47	http://www.adr.it/ - 43k -	47	your service
48	*Link* Cached -	48	www.touring-car.it
49	*Link* Similar pages	49	
50		50	*Link* Airport shuttle in Rome
51	*Link* ADR - Aeroporti di Roma Official Site	51	Door to door connection between the airports and
52	The company controlling Rome's two airports -	52	rome hotels.
53	Fiumicino and Rome Ciampino - provides airport	53	www.rome-airport-shuttle.com
54	maps and information on flights and facilities, ...	54	
55	http://www.adr.it/default.asp?L=3 - 42k -	55	**Heading level 2 Result Page:**
56	*Link* Cached -	56	
57	*Link* Similar pages	57	1
58		58	
59	*Link* ADR - Sito ufficiale Aeroporti di Roma - Error	59	*Link* 2
60	not found	60	*Link* 3
61	Sito ufficiale della Società Aeroporti di Roma.	61	*Link* 4

62	Informazioni, notizie su azienda, aeroporti di	62	*Link* 5
63	Fiumicino e Roma Ciampino, orario voli, arrivi e	63	*Link* 6
64	partenze, ...	64	*Link* 7
65	http://www.adr.it/0 - 19k -	65	*Link* 8
66	*Link* Cached -	66	*Link* 9
67	*Link* Similar pages	67	*Link* 10
68		68	*Link* Next **alt++**
69	… [other results]	69	
70		70	**Heading level 2 Search:**
71	Result page:	71	*Edit* airport rome **alt+c**
72		72	Search *Button*
73	First	73	Search:
74		74	*Radio button checked* the Web ***alt+w***
75	1	75	*Radio button not checked* pages from the UK ***alt+p***
76	*Link* 2	76	
77	*Link* 3	77	***Heading level 2 Advanced search:***
78	*Link* 4	78	*Link* Advanced search ***alt+a***
79	*Link* 5	79	*Link* Preferences
80	*Link* 6	80	
81	*Link* 7	81	**Navigation bar:**
82	*Link* 8	82	***Link Navigation help alt+h***
83	*Link* 9	83	*Link* Google Homepage ***alt+g***
84	*Link* 10	84	Web
85	*Link* Next	85	*Link* Images
86		86	*Link* Groups
87	*Edit* airport rome	87	*Link* News **alt+n**
88	Search *Button*	88	*Link* Froogle
89		89	*Link* more
90	*Link* Search within results |	90	
91	*Link* Language Tools |	91	**Search:**
92	*Link* Search Tips	92	*Edit* airport rome
93	*Link* Google Home -	93	Search *Button*
94	*Link* Advertising Programmes -	94	
95	*Link* About Google	95	**Heading level 2 Search Tools:**
96		96	*Link* Search within results |
97	copy2005 Google - Searching 8,168,684,336 web	97	*Link* Language Tools ***alt+1***|
98	pages	98	*Link* Search Tips
100		100	*Link* Google Home -
101		101	*Link* Advertising Programmes -
102		102	*Link* About Google
103		103	
104		104	Modified Google Interface - Searching 8,168,684,336
105		105	web pages

Figure 5: Google result page read by Jaws: the original page on the left, the modified one on the right

4.4 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.

As we see in the two scenarios, Roberto's and Sarah's experiences are very different. 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 6).

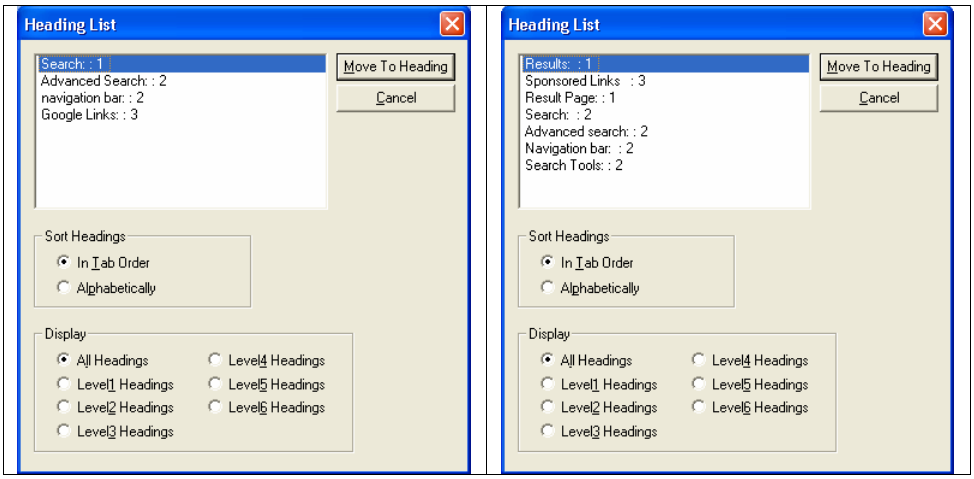

Figure 6: Logical sections of MGIs: a) home page and b) results page (generated by a specific Jaws command)

Thanks to the new structure of our proposal, 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 pressing 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 with 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. 2 and Fig. 5). On the home page of the original interface, when the user reads search fields and would like to find search options, they may 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 not be able to find them. This does not occur in the modified interface where the radio buttons come before the "advanced search" link (see Figure 2).
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 (Figure 5).
Actually, the repositioning of most important blocks of code in the first position in the interface source file had been the most relevant point noticed by users during a usability test aim at evaluating the interaction of totally blind users with the new and the original Google UIs [3].
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 to the user immediately, 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 will be able to interpret aural CSS properties this type of feature would be available with the appropriate style sheet.

5 Conclusions

In our study 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, colour, 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 jumping to

this point, and lastly, Tab keys and access keys for simplifying navigation around the interface. In addition we also moved 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 according to our design principles enable blind users to navigate the interface more quickly and efficiently, providing better orientation within the page content.

References

1. Andronico P., Buzzi M., Castillo, C., Leporini B. (2005). Improving Search Engine Interfaces for Blind Users: a Case Study, To appear in International Journal Universal Access in the Information Society.
2. Andronico P., Buzzi M., Leporini B. (2005). Increasing usability of search engine interface for visually impaired personas. In proc. Of the 3rd International Conference on Universal Access in Human-Computer Interaction (UAHCI), Las Vegas, Nevada, USA, 22-27 July 2005.
3. Andronico P., Buzzi M., Castillo, C., Leporini B. (2005). Search engine UIs: remote usability testing with blind persons. Technical Report IIT TR-15/2005, August 2005.
4. 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
5. Carroll John M. (1995) Scenario-Based Design, John Wiley and Son, USA.
6. 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
7. 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
8. 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.
9. 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.
10. 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
11. 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.
12. 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.
13. JAWS for Windows, Freedom Scientific, from http://www.freedomscientific.com
14. 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.
15. 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.
16. Mann, T.M. (1999). Visualization of WWW-Search Results. In In Proceedings of International Workshop on Database & Expert Systems Applications. pp. 264-268.
17. Nielsen, J., 2005. Mental Models For Search Are Getting Firmer, Alertbox, May 9, 2005 available at http://www.useit.com/alertbox/20050509.html

18. Nicolle, C., Abascal J. (2001). Inclusive design guidelines for HCI, p. 285, Taylor & Francis.
19. 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.
20. 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.
21. 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.