

Improving Search Engine Interfaces for Blind Users: a case study

PATRIZIA ANDRONICO*, MARINA BUZZI*, CARLOS CASTILLO**, BARBARA LEPORINI*

**Istituto di Informatica e Telematica del CNR*

***Center for Web Research, U. de Chile*

Abstract: This article describes a research project focused on improving search engine usability for the sightless persona, who uses assistive technology to navigate the Web. Based on a preliminary study on accessibility and usability of search tools, we formulate eight guidelines for designing search engine user interfaces. In the current phase of our study we modified the source code of Google's interface while maintaining the same “look&feel”, in order to demonstrate that with very little effort it is possible to make access easier, more efficient, and less frustrating for sightless individuals. After giving a general overview of the project, the paper focuses on interface design and discusses any implementation problems and solutions.

Keywords: accessibility, usability user interface, blind users, search engine

Introduction

Due to the enormous amount of information on the Internet today, search engines have become indispensable for finding specific and appropriate information. If we consider the user's task, two main search engine components are equally important to carry out a successful search:

- (I) the search process, which seeks the requested information and orders the results by relevance; and
- (II) the user interface, where the user types the query keywords and the search results are shown.

Since individuals interact with the search tool to set up a search task and explore the results, it is essential that user interfaces be easy to use and accessible to all; this is particularly important for sightless users who, interacting via screen reader, perceive the page contents very differently and experience a much longer search time.

In this study we describe a research project aimed at evaluating the accessibility and usability of several popular search tools available on the Web, in order to understand their limitations/drawbacks and propose improvements.

By means of automatic checking, gathering user feedback, and a screen reader, we were able to identify the main accessibility and usability problems. We then defined and proposed specific guidelines for improving search engine user interface design. Finally, by applying these directives to the UIs of Google, a search engine which is commonly considered accessible and easy to use, we were able to demonstrate that accessibility can be greatly improved while the interface maintains the same visual appeal. Specifically, we modified the Google interfaces by adding specific “hidden features” that greatly improve navigation and the search activity via screen reader, but do not affect the navigation of sighted users.

Background

Common problems encountered when navigating via screen reader are:

- *Lack of context* – The user accesses only a small portion of the text and may lose the overall context of the current page.
- *Information overload* – Portions of the site which do not vary (index, frames, banners) may overload the “reading” since the user hears the same items over and over for every page.
- *Keyboard navigation* – Since blind users do not use the mouse functions (i.e. pointing, scrolling, selecting, etc.) they move around the page using keyboard commands, such as Tab key, arrow keys, and so on.
- *Excessive sequencing in reading the information* – Basic screen reader commands for navigation and reading enable the user to explore page contents sequentially.
- *Screen reader interpretation* – The screen reader deals with Web page content in a very different way with respect to visual rendering. This requires a certain expertise in understanding advanced screen reader and browser commands, as well as orientation within the page itself, and both require considerable effort.

These drawbacks slow down navigation, annoying the user and provoking great frustration. Other user interface features should also be considered when improving usability:

- Component position. This is important since value-enhancing features are more "visible" when positioned in an area that is rapidly encountered by the eye and does not require page scrolling. In order for a sightless user to obtain a similar perception (i.e. visit the most important parts first) the main components of an interface should be located at the beginning of the page.
- Expressive power. A visual element communicates any kind of information much more rapidly and effectively than other methods. Keyboard navigation and sequential access through the page content is slow and can influence the navigation of blind users. Faster navigation and positioning over interface elements can be obtained by assigning access keys (keyboard shortcuts) and tab-index values (used when navigating sequentially) to the most relevant components. By means of shortcuts and "priority values" sightless users can quickly reach the desired search field or result link. Moreover, assigning a simple and familiar label to a field facilitates recognition via screen reader. Labels should be placed over or to the left of the field, to simplify exploration via arrow keys.
- Simple, clear design. It is very important to design a very simple interface in order for users to navigate the interface easily. Common design errors are unclear mapping of functions or too many functions grouped in the same space. This can require a non-intuitive understanding of the interface functions or an extra effort to remember the confused parts. We can observe this drawback in web directories whose interfaces are quite complex (full of elements) and can create confusion in a sighted user as well. A significant element in Google's success is related to its clear and simple user interface.
- Search criteria. A user typically performs a simple search and specifies one or more words, obtaining a large set of results. Advanced search options and commands can be specified to restrict search results, but, though powerful, are rarely used, even by skilled individuals. Instead, the function for searching within results (specifying additional keywords) can be easily used by unskilled users as well. However, due to its position at the end of result page and its font size and color, in Google it is not easy to see.
- Result exploration. Clustering permits users to explore results grouped by category, so users can navigate a single branch of results at a time, more

efficiently. If correctly implemented, this feature increases interface usability and could also save time finding pertinent results.

Although search engine companies have frequently attempted to improve usability, usually the design layout is aimed at sighted persons; thus in most cases, new and interesting UI features are useless for blind persons. Thus, although the Internet is a precious source of information and offers many services, these drawbacks can discourage visually-impaired users from attempting on-line access. In order to reduce this “digital divide”, the design of all interfaces should consider the special needs of the differently-abled. The individual as the center of any kind of artifact created to satisfy human needs is the main philosophy of User-Centered Design methodology (UCD). We can also assume that a user interface is a kind of artifact and should be designed with the users, their particular experience and cultural environment in mind. It is important for the designer/developer to take all these problems into account from the very first phase of the project/design.

Related Work

In the last few decades search technologies have evolved from dealing with small collections of homogenous and structured data, to indexing the enormous amount of heterogeneous and unstructured data available on the Internet today. Before, information was only available to a very skilled group of individuals such as researchers, librarians and information brokers (since a complex query language was necessary for interaction) while today anyone can look for anything on the Web, regardless of his/her training or expertise in information retrieval. This rapid, enormous expansion of the Internet led to the development of a new kind of search tool: the commercial search engine, such as Google, which provides free text searching of mostly unstructured data [Vaughan 03]. Popular Web servers, Web content providers, and e-commerce companies all employ search engines and offer interfaces to increase the “findability” of their information, products and services. For the Internet “at large” this is even more important: about 85% of users surveyed in [Kobayashi 00] use search engines and search services to find information. However, we have not yet arrived at very usable design solutions for search tool user interfaces, and even skilled individuals can encounter difficulties when making queries or seeking information. [Vaughan 03] reported that only 18% of users said they could find what they were looking for on the web. In

addition, 67% were frustrated when searching, 21% reported being able to find what they were looking for nearly every time, and only 60% reported finding relevant information most of the time.

Unfortunately, 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 gives modal access) actions take longer and tasks are more difficult since additional actions are required [Brajnik 04]. The gap between blind and sighted users' efficiency when performing online search tasks is explored in [Ivory 04]. In their 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.

Many studies focus on web interface accessibility and usability ([Scapin 00], [Nicolle 01]) but to our knowledge few involve the study of search engine interfaces. Research on search engines mainly addresses algorithms, strategies and architectures, and focuses on increasing the effectiveness and quality of results. For instance, regarding the relevance and precision of results, in [Agichtein 03] the authors introduce a meta-search engine which transforms a user's query into a set of more effective queries and re-ranks results on-fly, demonstrating that their system outperforms the underlying search engines.

Another branch of search engine research aims at improving the Graphical User Interface (GUI). An overview of the variety of possible visualizations for search engine results and a discussion of the main factors for their success can be found in [Mann 99]. In [Vegas03] 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).

Regarding accessibility, the W3C Consortium addressed this topic prior to 1997. The Web Accessibility Initiative Interest Group (WAI-IG) investigates the problems of accessing the web, presents alternative web browsing resources and produces guidelines for web content, authoring tools and user agent accessibility. Within the framework of the Web Accessibility Initiative, the W3 Consortium proposed a set of 14 main guidelines, called Web Content Accessibility Guidelines 1.0, <http://www.w3.org/TR/WCAG10/>, (presented in the Recommendation dated 5 May 1999) and is currently working in a 2.0 version

(<http://www.w3.org/TR/2004/WD-WCAG20-20040730/>). In addition, other accessibility guidelines have been defined, i.e. the Section 508 standards defined by the US Government (<http://www.section508.gov>). To facilitate testing of web content accessibility several automatic tools called validators have been implemented which automatically verify source code conformity to these guidelines.

Accessibility and usability requirements for sightless persons are discussed in [Leporini 02] and [Leporini 04], since both aspects are crucial for navigation via assistive devices. Another study concerning accessibility of on-line library resources for sightless persons was performed in [Schmetzke 02].

An interesting research focus on search engine design for the blind is presented in [Ferworn 00]. 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 involving the use of search engine, within the framework of the NOVA project (Non-Visual Access to the Digital Library) the Manchester Metropolitan University performed usability 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 [Craven03]. In [Ivory 04], a study analyzing the decision-making behavior and performance of blind and sighted users during the search task was conducted. This research aims to identify page features that could be presented in result 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 ability to specify criteria for controlling the order of results (ranking). Various ways to improve the user's search experience are suggested.

Research Objectives

This study considers accessibility and usability issues related to interacting via screen reader, in order to propose new solutions for improving the search engine interface and providing easier access. Our investigation consisted of four steps:

- i. Identifying the main accessibility issues.**

Automatic heuristic checking of accessibility. We selected the most popular search tools in Italy, and others that display interesting features

either in the search interface or in the presentation of results, as discussed in the next section. Using two automatic evaluators we checked all the search tool interfaces to detect the most important technical problems, according to W3C standards.

ii. Investigating usability issues.

Empirical exploratory research. A questionnaire was prepared in order to learn how individuals use the search tools, their satisfaction, problems, suggestions for improving UIs and so on.

Data analysis. The data collected from this automatic heuristic checking and from the empirical research were then analyzed to produce a preliminary evaluation report.

iii. Proposing specific guidelines.

Understanding interaction via screen reader. We chose Google for our study, since it is the most popular search engine used in the Italian sightless community. Specifically, we highlighted problems that may arise when interacting via screen reader, by simulating the execution of a simple search task on Google.

Definition of specific design guidelines. Based on what we learned from our study, we proposed a list of general principles for UI designers to follow.

iv. Proposing and testing a possible solution.

Improving Google interfaces. At this point we developed modified Google interfaces, following our guidelines for taking into account the needs of blind users. We did not modify the original graphic style of the interface (i.e. the visual layout) but only modified the source code for better navigation via screen reader.

Testing the proposed solutions. In order to evaluate the effects of the proposed criteria and the new user interfaces, we plan to prepare a user test involving sighted as well as visually-impaired users. During the test we will assign several tasks to participants in order to evaluate the effects of our proposed principles.

The first three steps are discussed in detail in [Andronico 04], [Buzzi 04], [Leporini 04b]. We briefly describe the most important results in the next three chapters, and then introduce a possible solution in the next section.

Identifying the main accessibility issues

We began our research with an automatic validation test of the selected search tools in order to verify whether their interfaces conformed to the W3C accessibility guidelines. For cross- testing, we selected two free validators: Bobby¹ (<http://bobby.cast.org/>) and Torquemada² (<http://www.webxtutti.it/>). We observed that Bobby conveniently arranges errors by priority level as defined in WCAG 1.0³, while Torquemada lists errors in the same order they appear in the source code. Furthermore, Bobby's report is more complete than that of Torquemada, which is still under development. As previously mentioned, we chose seven search tools that were either the most popular or presented a particular feature:

- **Google** (<http://www.google.com/>), and **Altavista** (<http://www.altavista.com/>), very popular search engines;
- **Yahoo** (<http://www.yahoo.com/>) and **Excite** (<http://www.excite.com/>) web directories and meta-searches. Yahoo has its own search engine.
- **HotBot** is a meta-search which permits customizing the user interface (<http://www.hotbot.com/>);
- **Vivisimo** is a meta-search which performs on-fly clustering of results (<http://vivisimo.com/>);
- **Kartoo** is a meta-search which displays results with a series of interactive maps (<http://www.kartoo.net/>).

The tests were performed on four types of interface⁴ for each search tool: home page (simple search), advanced search, preferences and results.

Due to the nature of accessibility guidelines, which require human supervision to be efficiently evaluated, (for example, some guidelines concern pictures or color contrast between background and foreground) we manually checked the validator output.

¹ "Bobby is a web accessibility desktop testing tool designed for small websites to help expose barriers to accessibility and encourage compliance with existing accessibility guidelines, including Section 508 of the US Rehabilitation Act and the W3C's Web Content Accessibility Guidelines (WCAG), on a page-by-page basis." From <http://www.w3.org/WAI/ER/existingtools.html>

² "Torquemada is an Italian-language tool designed to assess Website accessibility. The initial version is an online service, but the tool is under development and a downloadable version is promised soon." From <http://www.w3.org/WAI/ER/existingtools.html>

³ WCAG: Web Content Accessibility Guidelines 1.0, W3C Recommendation 5 May 1999. <http://www.w3.org/TR/WAI-WEBCONTENT/>

⁴ The test on the interfaces, performed during a degree thesis, has been completed in October 2003.

Of all the tools, only Google conformed to priority 1 of WCAG 1.0, meaning that it satisfies the minimum accessibility degree (level A), whereas other search engines, directories and meta-searches presented priority 1 errors.

The analysis showed that errors are recurrent: different search engines present the same defects. For instance, tables are frequently used for the layout of page results, ignoring the needs of sightless individuals for whom table contents are very difficult to access due to the sequential reading. In some interfaces, one type of error is present in one part of the page source but not in another. This suggests that various updates have been performed in the past by different tools/persons, resulting in inconsistent attention to accessibility.

Investigating usability issues

To gather user feedback, we drew up a four-part questionnaire comprising: user characterisation, general knowledge of search tools, use of search interfaces (simple/advanced search and preferences). The information was collected and used in absolute confidentiality.

The questionnaire was distributed to 52 individuals. The sample consisted of 75% sighted and 25% sightless users; 33% were women and 67% men; age range was from 20 – 60+ years. Of this sample 54% were ICT-skilled; however, all subjects use the computer either at home (13%), work (44%) or both (43%). Google was the most popular search engine, followed by Altavista and Yahoo. Our first observation was that sightless users do not use web directories. For the blind, an interface crowded with elements is very difficult due to the more complex visual layout and structure, and it may become impracticable.

Feedback on knowledge of search tools showed that 60% of users always used the same search engine but 79% have tried performing queries with different search products. This value may indicate that the user attempts to find better user interfaces. In addition, 58% utilize search engines frequently while 33% do so only when needed.

The third part of the questionnaire concerned how users utilize search engines. Only 25% of the subjects had attempted to configure the search tool (i.e. the preference page). However 75% have used the advanced search; of these 38% were blind and 87% sighted users. This data confirms that interaction with a more complex interface is more difficult for blind users. About 87% of the total sample (blind and sighted users) agree that using a search engine is the fastest way to find

information on the Internet. In all the following figures the diagonal-row pattern represents sighted users, while the solid color indicates blind users.

Usability renders Internet navigation more effective, efficient and satisfying. To the question “do you think that search engines are easy to use?” 92% of sighted users answered “yes”; in contrast, 77% of blind users said “not always”, as shown in Figure 3b). This result highlights the fact that usability is crucial for disabled persons.

Concerning queries, users usually specified more than one keyword (92% of sighted users and 69% of blind users) as shown in Figure 1a. Furthermore, 67% of sighted users had no difficulty choosing the right keywords for the query, whereas only 38% of blind users agreed with this.

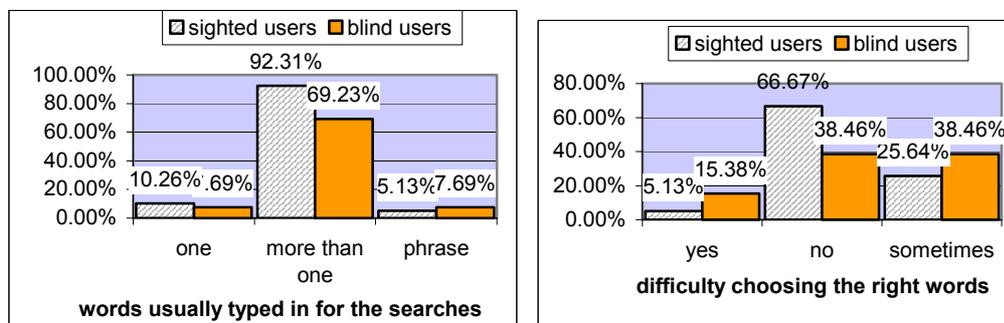


Fig. 1. - a) number of keywords b) difficulty choosing the keywords

Regarding results, 67% of sighted users explored more than two pages compared to 15% of blind users, while 80 % of blind users accessed only the first two (Fig. 2a). Once again these data reflect the difficulties of sightless users. The ability of sighted users to rapidly focus on interesting results or discard irrelevant information is greatly reduced in blind users due to the sequential access to page contents. Thus the blind need more time to visit each result page and accessing more than two pages becomes difficult, if options for rapid navigation via keyboard are not offered.

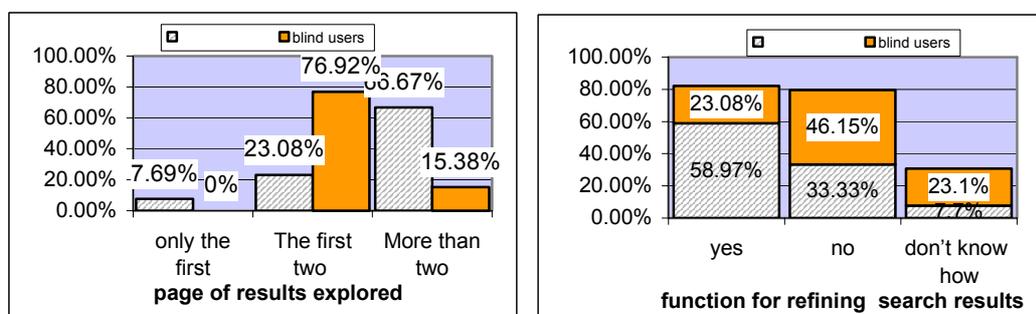


Fig. 2. - a) page of results visited b) searching in results

In addition, only 23% of blind users use the refining function (for searching into results) compared to 59% of sighted users (Fig. 2b). Lastly, sponsored results

were known to 48% of the whole population, but only 25% were able to recognize them among all results.

The last part of the questionnaire attempted to identify the greatest difficulties for users. This question permitted multiple answers; Figure 3a shows the results. For sighted users the main obstacle is choosing the right keywords (62%) while blind users also have difficulty reading results (46% compared to 15% of sighted users) and accessing interfaces (functions/interfaces unclear): 31% compared to 18% of sighted users. Lastly, 90% of sighted users nearly always find what they are looking for, while 38% of blind users find useful information only sometimes and 8% almost never! (Fig. 3b).

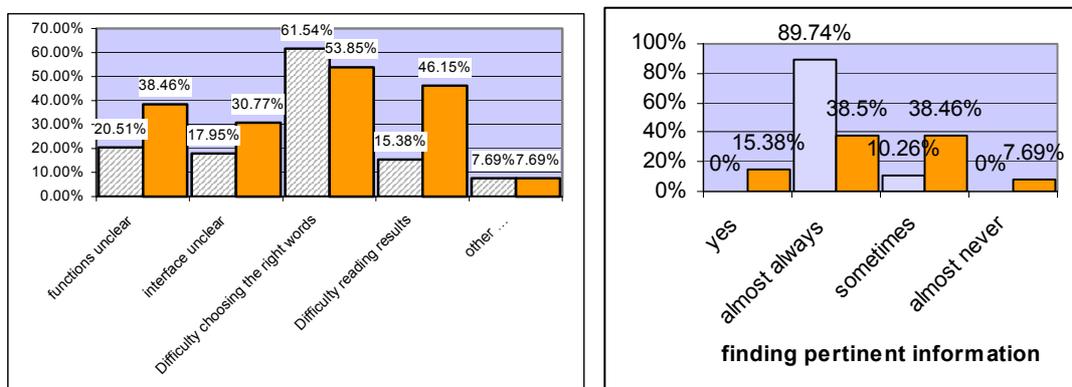


Fig. 3. – a) Difficulties for users b) finding what you are looking for

These data demonstrate that sightless individuals are more sensitive to user interface design. Thus, it is crucial to structure results in a way that permits blind users to access them quickly and easily. For instance, reaching the result area rapidly, clearly knowing the number of results obtained, reading appropriate links about the results, and so on, are important features for users who access information by means of a voice synthesizer or Braille display.

Proposed guidelines

For users interacting by means of any kind of assistive technologies, the UI layout and structure are crucial, since when navigating by screen reader the user perceives page content in a very different way from its rendering on the screen. Originally the HTML language was developed for structural markup of a document, but in today's Web, it is still used for formatting. The developer should be aware of how the screen reader handles Web page layout, and how blind or visually-impaired users perceive page content and interact with the interface. The main issues in UI design are:

- A. **Page content serialization.** The JAWS screen reader takes the page source and serializes its content (link, edit field, button, cell, and so on). Also frames or blocks <div> are lined up, without taking into account specific positions assigned by CSS properties. Basically, JAWS reads the code as it was written and lines up the page content in the form of a single column. Thus, the order in which the blocks <div> and the frames are written is very important.
- B. **Navigation by Tab key and special commands.** It is important to remember that a blind user usually prefers to visit the page link by link (by Tab key) or use special commands in order to move quickly around the pages. Hence, it is important to facilitate navigation via keyboard by assigning a scale of importance to the links, applying shortcuts to main elements, using specific tags such as <Hn>, and so on. Furthermore, many special screen reader commands operate well only if the developer has applied specific tags or attributes, or appropriate criteria have been followed.
- C. **Differences between visual layout and aural perception.** Often when developers design a Web page they provide some useful information by means of visual features, such as position, color, separating blank spaces, formatting features, and so forth. For instance, some secondary information is placed on the side so that users can recognize it immediately. It is important to provide the same “message” to a blind user by another means (e.g. using a table, a heading, a hidden label, etc.).

Taking into consideration the above issues as well as accessibility and usability difficulties resulting from the screen reader, we propose that the following principles be considered when designing a search engine interface layout:

1. **Easy location and labelling 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. Be careful to correctly match <label for> with input elements, and put labels above or to the left of the input element, rather than below.
2. **Highlighting the search result.** Use a heading level (i.e. <h1> or <h2>...<h6>) at 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 the 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.** 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 `` or `` elements; each item on the list should consist of a single result. This feature enables the screen reader to inform the user of the number of items; the user is then able to skip quickly item by item. The page should contain a maximum of ten items.
4. **Recognizing sponsored links.** Keep sponsored links separate from the other results. Thus, put them in a clearly labeled separated 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.** Locate the 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 the pointers to the next 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. **Navigating more quickly.** Assign a scale of importance (i.e. by the tab index attribute) so users can reach the most important elements quickly. In 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.
7. **Alerting by sound.** Different sounds for different events should provide useful information for blind users. For instance, two different sounds may be used to indicate the success (at least one result) or failure (no result) of the search. However, a more complex sound assignment could be applied.
8. **CSS2 aural style sheets.** 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.

Proposing and Testing a Possible Solution

In the last step, we decided to apply our proposed guidelines to a specific case. Since Google was the search tool preferred by blind people (in our study), we chose its user interface. Specifically, we modified the original Google interface (source code) for simple search and result exploration, while carefully maintaining the original graphic layout. In the following, we refer to these new interfaces as “modified Google User Interfaces”. By means of the new interface design we intended to map the visual functions as aural-usable information, in order to achieve rapid positioning in the desired interface area, immediate information on search status, etc. Since we intend to show how the user interface can be improved while keeping the same visual layout, the changes only involve screen reader interaction; if the needs of visually-impaired people are also considered, changes will affect the visual layout as well.

We will describe the technical approach to this part in detail in the following section.

Reengineering of Google UIs

In the current phase of our research, the aforementioned guidelines were applied to the Google GUI in order to shorten the time needed to carry out a search task. Our idea is to re-engineer the code of Google web search pages without changing the original “look&feel” but facilitating interaction by screen reader.

Reengineering the user interfaces required a complete redesign process, but under the constraint of an unmodified visual layout. To accomplish this, we separated the page’s information from the visual rendering, using the cascading style sheet (CSS), one of the W3C Recommendations, that permits to organize the page into logical blocks. Thanks to the main characteristics of the CSS language, each block can be positioned in the XHTML code without changing its position in the graphic layout. For this reason our first steps were an attempt to re-write the code in accordance with the last standard on the web:

- Structuring the page content in logical sections;
- Modifying the order of the functional block of elements in the page (e.g., navigation bar, search fields, search options, result list, etc.).

Furthermore, we implemented some features that make the interface more accessible in general and simpler to navigate via screen reader:

- Building a different navigation order when interacting through tab index or access key;
- Specifying information useful for the screen reader only (i.e. hidden labels);
- Adding aural feedback (i.e. specific sounds for different events).

For this investigation we considered Internet Explorer, since the Jaws screen reader is tested with this browser and does not work properly with other ones.

Google offers two slightly different versions:

- the .com, the original one;
- the regional versions, such as .it, .uk and .fr. This version furnishes options (radio buttons) to restrict the search to the specific domain (it, .uk, .fr.) or language of the region (if it is different from English).

Since we need English for writing up our results and Italian for conducting the usability test, we decided to work in parallel with the Italian (<http://www.google.it/>) and the UK (<http://www.google.co.uk/>) versions.

Restructuring the code

One of the most frequent problems encountered by a sightless person navigating via screen reader is misunderstanding or losing track of the content. This negative effect is due to the difference between the visual layout and perception via screen reader. In particular, the use of a table for obtaining a graceful rendering may cause different parts of the text to be out of order in the sequential reading. Also, Google uses tables just for layout purpose both in the simple and in the advanced search UIs. For this reason, our first aim was to demonstrate that with very little effort, by eliminating the tables, it is also possible to structure elements in an interface as simple as Google.

Re-write the code according to the latest standards

Figure 4 shows the original (a) and modified (b) Google Home Page loaded in the Internet Explorer browser. At first the two interfaces look quite the same; instead, their interpretation by Jaws is very different (as shown in Figure 9).

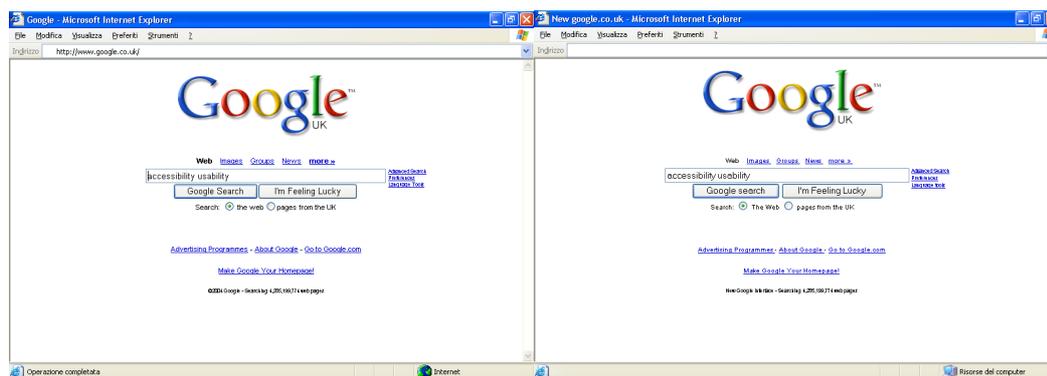


Figure 4 – a) Google home page available at <http://www.google.co.uk> b) Our modified interface

Looking more closely at the source code of the original Google UI, we observe that two tables have been used, which are invisible on the screen since the cell border was set to 0 pixels. Figure 5 shows the cells of the tables used in the layout of the original Google Home Page.

In the result page thirteen tables are used; some of them are nested. In this case, the effort required of a blind person increases greatly due to the fact that she/he needs to keep in mind the more complex logical sequence of the page structure that she/he cannot see.



Figure 5 – Use of tables in the UK Google home page

To refine Google user interfaces, we then decided to follow the same logical structure as the graphical layout which divided the entire contents into four different sections (highlighted in red in Figure 6):

- 1) Navigation bar
- 2) Search box and options
- 3) Advanced Search and Preferences
- 4) Google info and other links (Google Links)



Figure 6 – Logical sections of the modified Google home page

In order to define those four sections, we considered the CSS block properties rather than the layout tables. In practice, we used <DIV> tags for structuring the content and elements. When the graphic interface is visually perceived, it is easy to identify the four main sections based on the arrangement of elements. Our goal is to give a similar opportunity to users navigating by screen reader. Since we based content structuring on <DIV> blocks, a preliminary solution might consist in giving an appropriate “title” attribute to the <DIV> blocks which embody the four main sections (for example: div class=“navbar” title=“navigation bar:”). Unfortunately, at present (as of the writing of this paper) the latest version of the screen reader used for our test (i.e. Jaws 5.10) is unable to interpret the “title”

attribute assigned to the <DIV> element (i.e. the <DIV> title is not announced); therefore, an alternative solution is proposed. If the screen reader makes this feature available, developers should define a meaningful “title” for the main blocks of the page so that the user, once he has the “title” attribute list, can quickly learn the page structure and easily jump between sections.

We used the same technique for the results page and divided it into eight sections as shown in Figure 7.

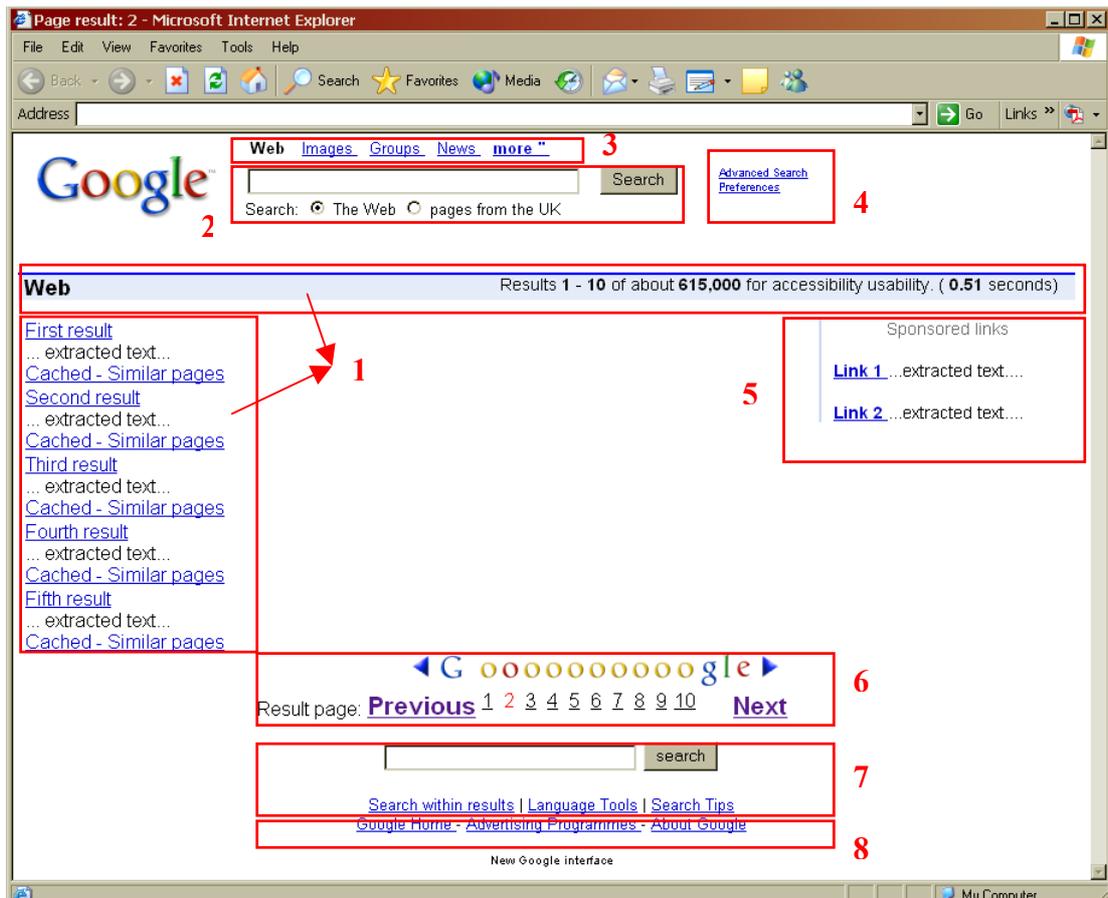


Figure 7 – Modified Google interface: results page

One of the features provided by Jaws is the possibility of recognizing the heading levels - i.e. content enclosed between <Hi> tags and the corresponding heading number are announced - and jumping to another heading level within the page. Since <h1>...<h6> tags are an efficient and encouraging way to achieve good document structure (e.g., chapters, paragraphs, and so on), we propose their utilization for structuring the web interface. The idea is to assign a heading level <Hi> to the “title” (in our case a hidden label) of each page section. Practically, this approach “extends” the heading level usage. Our suggestion is to apply heading levels for good page structure, whether it is a document or any generic Web content. thus, the page content is logically structured into several sections.

Each section should have a “title” either visible on the page or masked by an appropriate hidden label. In this method, an “index” of the sections available for the page can be created on- fly (i.e. by a special Jaws command “Insert+F6); then, moving to a desired section, or navigating among the headings (previous and next), is quite easy.

Figure 8 shows the Heading Lists of the Google modified interfaces (home page and result page), generated by Jaws. We defined four headings for the modified home page and eight for the result page. Note that the original Google interface does not present such a structure. Three different importance levels were assigned: “<H1>” for the most important page section; “<H2>” for sections which might be useful; and lastly “<H3>” for sections that are not particularly useful. For instance, in the Google home page the most important section is “Searching for”, while for the result page it is “Search results”

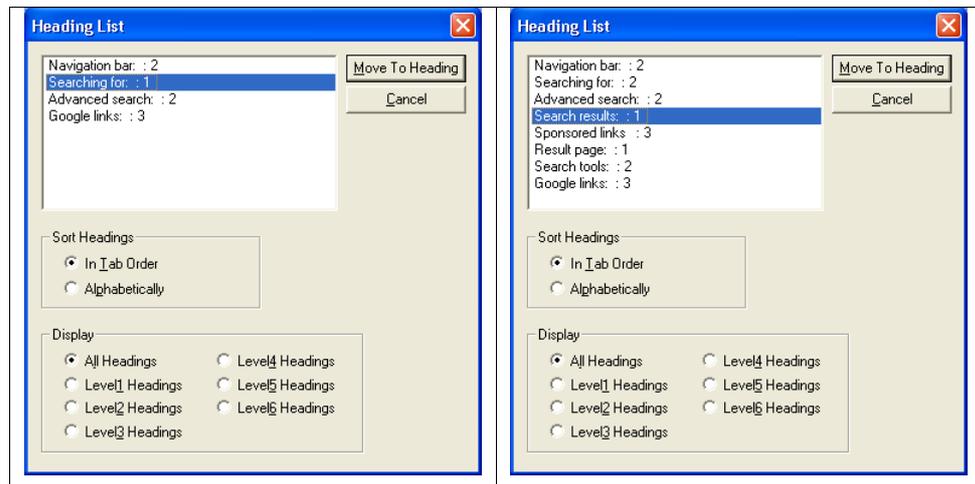


Figure 8 – Logical sections of the modified UIs: a) home page and b) results page (generated by a specific Jaws command)

Thanks to this 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 heading list. With the arrow keys he/she can select the desired section and just push the Enter key.
- The “h” and “shift+h” commands permit users to jump to the next or previous heading (independently of its level); in this way users move sequentially within interface sections.
- By pressing the “1”, “2” ... “6” keys, users jump directly to the next (or prior if shift is associated) Heading at Level. 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. In any case, other specific commands for skipping to the first or last heading are available.

New logical order

Once we decided which sections to consider on the page, we defined the CSS file to specify the rendering options and position each section on the original part of the screen. Each section was then considered as a unique block according to the CSS language, with the `<div>` property. The `<div>` tag associated with a specific identifier permits us to order the element in a position in the code that may be different from its position on the screen. For example, the first element in a GUI⁵ unnecessary is the first element in the XHTML code.

The order of the blocks is quite important, both for a correct interpretation of the content, and to save time when reading the page sequentially. In fact, the screen reader reads the `<div>` blocks as they are written in the page code. Originally, we planned to organize the code by inserting elements according to their degree of importance (from the most to the least relevant) and using the CSS position to properly visualize the element blocks on the screen. Unfortunately, we were unable to use the absolute positioning property of the CSS, since some browsers, such as Internet Explorer v. 6 (the browser used in our investigation) still do not support this feature. For this reason, we compromised between the best technical solution and one that is actually applicable, due to the current state of screen reader and browser compliance to standards.

Figure 9 reports the source code in both the original (a) and the modified (b) Google home page. Although the source code is increased, a blind user can easily jump to the desired section, or navigate sequentially after visiting the most important sections first.

⁵ We assume that the first element on the screen is the first element encountered by our eyes when we read or scan a page (according to our culture, left-right and top-bottom).

<p>Graphic Google</p> <p>Web</p> <p><i>Link</i> Images</p> <p><i>Link</i> Groups</p> <p><i>Link</i> News</p> <p><i>Link</i> more »</p> <p><i>Edit</i></p> <p>Google Search Button</p> <p>I'm Feeling Lucky Button</p> <p><i>Link</i> Advanced Search</p> <p><i>Link</i> Preferences</p> <p><i>Link</i> Language Tools</p> <p>Search:</p> <p><i>Radio button checked</i> the web 1 of 2</p> <p><i>Radio button not checked</i> pages from the UK 2 of 2</p> <p><i>Link</i> Advertising Programmes -</p> <p>...</p>	<p><i>Graphic Google logo</i></p> <p>Heading level 2 Navigation bar:</p> <p>Link Navigation help alt+h</p> <p>Web</p> <p><i>Link</i> Images</p> <p><i>Link</i> Groups</p> <p><i>Link</i> News alt+n</p> <p><i>Link</i> more »</p> <p>Heading level 1 Searching for:</p> <p><i>Edit alt+0</i></p> <p>Google Search <i>Button</i></p> <p>I'm Feeling Lucky <i>Button</i></p> <p>Search:</p> <p><i>Radio button checked</i> The Web alt+w 1 of 2</p> <p><i>Radio button not checked</i> pages from the UK alt+p 2 of 2</p> <p>Heading level 2 Advanced Search:</p> <p><i>Link</i> Advanced Search alt+a</p> <p><i>Link</i> Preferences</p> <p><i>Link</i> Language tools alt+l</p> <p>Heading level 2 Google Info:</p> <p><i>Link</i> Advertising Programmes -</p> <p>...</p>
---	--

Figure 10 – Screen reader interpretation of the: a) original, and b) the modified UK Google home page

Additional features

Once we finished restyling the page, we decided to improve accessibility and usability characteristics by adding some additional features, explained in detail in the following paragraphs.

Quick navigation

In order to improve navigation among the elements in the sections and by keyboard, it is advisable to use tabbing order and shortcuts. For this reason, we had assigned some tab-index and access keys to two pages: the home page and the result page. The access keys were chosen to correlate with the associated element that was the most intuitive for a user. For example we chose the letter H for navigation help, or G for Google home page. Table 1 summarizes all the shortcuts introduced in the modified Google UI.

Key	Description
H	Navigation help
G	Google home page
O	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

Table 1 - Shortcuts in the modified Google UIs

We applied different levels of tabbing order for the home page and result interface. In the home page, the search field and options are more relevant than other parts; in the results page we thought that the number of results and the result links should be visited before other elements. For this reason, different behavior occurs when the user presses the Tab key in those pages. The next table summarizes the visiting order when the user moves by Tab key in both the Google home page and result page.

Search box and option buttons; Search push buttons (i.e. "Google search" and "I'm Feeling Lucky"); Navigation Bar; Advanced Search and Preferences; Google Info and other links.	Search results (i.e. Results 1 - 10 of about...); First result, Second result, etc. (cached and similar links are skipped); Result pages (Prev, 1, 2,...); Search Tools (i.e. "Search within results" and "Search Tips"); Sponsored Links; Searching for; Advanced Search Navigation bar; Google Info and other links; Cached and similar page links.
--	--

Table 2 – Visiting order with Tab Key navigation: the modified (a) home page, and (b) result page

Figure 11 shows the code used for specifying tab index levels and shortcuts. The parts in bold refers to the tags for tabbing order and those in italics refer to the shortcuts. A lower tab-index value indicates greater importance (i.e. the element

with `tabindex=1` is the most important). Interacting elements without tab index are visited for the last ones (e.g. cached and similar pages).

Notice in the figure the use of the tag ``. This tag is defined to create a false link, which allows the users pressing the Tab Key to jump immediately to “Result 1 of...”. The hidden labels are described in the next section.

```
<a name="results" tabindex="1">
Results 1 - 10 of about </a>
...
<div id="result-list">
<div class="r">
<p class="hidden-label"> 1. </p>
<a href="..." tabindex="1"> First result </a> <br />
...extracted text... <br />
<a href="..."> Cached </a> -
<a href="..."> Similar pages </a>
</div>
<div class="r">
<p class="hidden-label"> 2. </p>
<a href="..." tabindex="1"> Second result </a>
... </div>

<div class="nav-pages">
<a href="..." accesskey="-" tabindex="5">  <br />
<strong>Previous</strong></a> </div>
...
<div class="nav-pages">
<a href="" accesskey="+" tabindex="5">  <br />
<strong>Next</strong></a> </div>
```

Figure 11 - Fragment of result page code containing tab index and access key assignments

Hidden labels

“Hidden labels” are another feature we added in order to simplify navigation for a blind user. In this case, a label added to a particular section is hidden to a sighted user, but is read by the screen reader. Like other options provided by the CSS language, such as the ALT attribute for an image, or the TITLE attribute, our

labels aim to facilitate recognition of an element in a page, whereas no attribute does this.

We used the hidden labels for the following purposes:

- Marking the search edit fields (e.g. “searching for”);
- Marking interface sections (“navigation bar”, “search results”, etc.);
- Inserting blank lines to separate content blocks and to make reading via screen reader clearer.

The “key” for defining our hidden label is the use of the z-index css property which means the hidden label is placed in the layer -1, i.e. under the basic layer (z-index=0) containing the page. In order to assure the label’s invisibility on the screen it is not possible to use the *display:none* and *visibility: hidden* properties since the screen reader interprets these directives and does not read the label.

Furthermore, at this time, the *media:aural* property is not supported by either the Jaws screen reader or the IE browser. Thus, to guarantee the label’s invisibility, a solid color should be assigned to the DIV block covering the hidden label.

Aural feedback

Aural feedback is very useful for blind users since it permits them to associate a specific sound with a given situation. A sightless user is used to configuring the operating system and its applications so that specific sounds are reproduced when certain events occur. We can call this a “sound icon”, since each aural message has a specific meaning.

We then applied this feature to the Google interfaces, adding several aural icons in appropriate, specific positions, such as:

- o When the edit field receives the focus;
- o When the user types the search text;
- o When a radio button for setting the search option (i.e. “the web” or “page coming from ...”) is selected: lower tones are used;
- o A different sound for the success or failure of the search process.

The last case is especially useful for a blind user, because it immediately informs him whether the search has succeeded, without having to explore the page content.

In order to link a sound to a specific event or situation, the best way would be to use aural css properties. Since current browsers do not recognise and interpret aural css properties, our interface uses javascript, as shown in Figure 12.

```

//Definition of events-file of sounds
var snd= new Array();
snd[0] = sndpath+"pageload.mid";
snd[1] = sndpath+"ontextf.wav";
snd[2] = sndpath+"key.mid";
snd[3] = sndpath+"sradiob1.mid";
snd[4] = sndpath+"sradiob2.mid";
snd[5] = sndpath+"sradiob3.mid";
snd[6] = sndpath+"results.mid";
snd[7] = sndpath+"noresults.wav";

function playsound(sound)
{
    document.all.music.src=snd[sound];
}

```

Figure 12 - javascript code: association of sounds with events and play function

To customize the interface, a sound option should be available for enabling/disabling sounds.

Implementation

Although we could have used static, simulated pages for the search engine's results, to carry out user testing we needed to implement the modified Google interface with “live” results. For this, we needed to query Google, parse the result set, and re-format them, applying the guidelines described above.

Fortunately, Google provides programmatic access to its search results using the “Google API” (<http://www.google.com/apis/>), which is specified using the Web Services Description Language (WSDL).

In short, WSDL describes a set of messages which can be exchanged between a client and a server. In this case, the client is our modified interface and the server is the search engine. The messages are encoded in XML and passed using the Simple Object Access Protocol (SOAP). This protocol enables the exchange of structured and typed information (messages) between peers in a decentralized, distributed environment, by using a variety of "underlying" protocols, including HTTP (as in the case of our application), SMTP (SOAP over E-mail), RPC.

A SOAP message is XML-based information composed of two elements enclosed in an envelope: i.e., a header (optional) and a body, as shown in Figure 13. The response message to a search query looks like this XML document:

```

<?xml version='1.0'?>
<SOAP-ENV:Envelope> <SOAP-ENV:Body>
  <return>
  <resultElements>
    <item>
      <URL>http://www.auto-europe.co.uk/guides/Italy/Pisa-guide.cfm</URL>
      <snippet>... Each of these towns had both a merchant ...</snippet>
      <title>Rent a Car in Pisa from Auto Europe Car Rentals</title>
    </item>
    <item>
      <URL>http://travel.yahoo.com/p-travelguide-3678607-pisa_pisa-i</URL>
      <snippet> ... City of Pisa ...</snippet>
      <title>City of Pisa - Yahoo Travel</title>
    </item>
    ...
  
```

Figure 13 - extract of the response to a query from the Google APIs. Details such as data types and name spaces are omitted for clarity

As we can see, the results are structured simply. Each single result of the query (the `<item>`) has a description (`<snippet>` element), URL and title.

After the response is received from Google, it has to be transformed into an XHTML page. To do this, we use the XSLT (eXtensible Stylesheet Language Transformations) language, which described the transformation of a source XML document into a destination XML document. In our case, the source XML document is the SOAP response received from Google, and the destination XML document is an XHTML document describing the results interface.

```

<xsl:template match="resultElements">
  <xsl:if test=".item"><p><b>Web page results:</b></p></xsl:if>
  <ol><xsl:apply-templates select=".item"></ol>
</xsl:template>
<xsl:template match=".item">
  <i>
    <xsl:element name="a">
      <xsl:attribute name="href"><xsl:value-of select=".URL"></xsl:attribute>
      <em><xsl:value-of select=".title"></em> <br/>
    </xsl:element>
  </i>
</xsl:template>

```

Figure 14 - extract of a simple XSLT stylesheet for transforming results. Elements from the response are underlined

XSLT provides a flexible way to modify results and test different solutions. Moreover, in some browsers, such as Mozilla, alternative XSLT transformations can be provided by the Web server and executed by the browser, providing different rendering for different users. However, in our case we are interested in providing a single interface for all users, with the same layout but hidden markup for improving accessibility and usability.

Unfortunately the index used by Google APIs is different from (smaller than) those used by Google .COM and regional versions, since they are intended for research use only. Thus results of the same query do not produce the same results when the user sends a query using the browser and via APIs. For these reasons, in order to set up the environment for user testing we needed to create two transformations, one re-creating the original Google Interface and the other for the modified Google Interface.

Conclusion and Further Work

Our analysis aims to demonstrate that usability for blind users can be greatly improved while maintaining an appealing graphic layout. To accomplish this, we redesigned the Google user interfaces by taking into account problems of navigation via screen reader.

First, we structured the UI in logical sections, grouping interface elements by function. Interaction via screen reader was then simplified. Specifically, we used heading levels to assign different degrees of relevance to different parts of the interface (i.e. logical sections), and by using “hidden labels” we marked these sections to be perceived only by screen reader. Then aural feedback was added to further simplify the interaction.

In the future, in order to evaluate the effects of the proposed criteria and the new user interfaces, we plan to make a specific user test with sightless users. During the test we will assign several tasks to participants in order to actually evaluate the effects of our proposed principles. We wish to demonstrate that a blind user can perform fewer steps to accomplish a specific search task using our interface instead of the original one.

In conclusion, redesigning an existing site can be onerous in the case of large, dynamic sites, but for search engines, which have at most four interfaces (simple search, advanced search, results and preferences) the cost is low and benefits are considerable for any user.

References

- [Agichtein 03] Agichtein, E., Lawrence, S., Gravano, L., (2003), Learning to Find Answers to Questions. In *ACM Transactions on Internet Technology (TOIT)*, Volume 4, Issue 2 (May 2004) pp. 124-162.
- [Andronico 04] Andronico P., Buzzi M., Leporini B. (2004) Can I find what I'm looking for? Alternate Track Papers & Posters the Thirteenth International World Wide Web Conference NY, USA, pp. 430-431.
- [Buzzi 04] Buzzi M., Andronico P., Leporini B. (2004) Accessibility and Usability of Search Engine Interfaces: Preliminary Testing. Adjunct Proc. of 8th ERCIM UI4ALL Workshop, Vienna, Austria.
<http://www.ui4all.gr/workshop2004/publications/adjunct-proceedings.html>
- [Brajnik 04] Brajnik G. (2004) Achieving Universal Web Access through Specialized User Interfaces. Adjunct Proc. of 8th ERCIM UI4ALL Workshop. Vienna, Austria.
- [Craven03] Craven, J., Brophy, P. Non-visual access to the digital library: the use of digital library interfaces by blind and visually impaired people. Tech. report, Manchester: Centre for Research in Library and Information Management.
- [Damery 03] Damery E. (2003) JAWS for Windows version 4.51. New features and accessing the Internet. Proceedings of 18th Technology and Persons with Disabilities Conference (CSUN), Los Angeles, California, USA.
- [Ferworn 00] Ferworn, A., Bodner, R., Chignell, M. H., Auditory WWW Search Tools. In *Int. Conference on Auditory Display 2002*. Atlanta, USA, April 2-5, 2000.
<http://www.icad.org/websiteV2.0/Conferences/ICAD2000/ICAD2000.html>
- [Ivory 04] Melody Y. Ivory, Shiqing Yu, Kathryn Gronemyer. Search result exploration: a preliminary study of blind and sighted users' decision making and performance. CHI 2004 Extended Abstracts, pp. 1453-1456.
- [JAWS] JAWS for Windows, Freedom Scientific. <http://www.freedomscientific.com>
- [Kobayashi 00] Kobayashi, M. and Takeda, K. (2000). Information retrieval on the Web. *ACM Computing Surveys* Vol. 32 (2000), pp. 144- 173.
- [Leporini 02] Leporini, B., Paternò, F. Criteria for Usability of Accessible Web Sites. In *User Interfaces for All 2002*, Paris (Chantilly), FR, 23-25 Oct 2002, pp. 43-55.
- [Leporini 04a] Leporini, B., Paternò, F. Increasing 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 04b] 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 99] Mann, T.M. (1999), Visualization of WWW-Search Results. In *International Workshop on Database & Expert Systems Applications*. pp. 264-268.
- [Marti 99] Marti A. Hearst (1999), User Interfaces and Visualization, In *Modern Information Retrieval* by Ricardo Baeza-Yates and Berthier Ribeiro-Neto, by Addison Wesley, pp.257-323.
- [Nicolle 01] Nicolle, C., Abascal J. Inclusive design guidelines for HCI, p. 285, Taylor & Francis, 2001.
- [Scapin 00] Scapin, D.L., and al. In *Conf. on Human Factors and the Web*. Austin, Texas, 19 June 2000.
<http://www.tri.sbc.com/hfweb/scapin/Scapin.html>
- [Schmetzke 02] Schmetzke, A. The Accessibility of Online Library Resources for People with Print Disabilities: Research and Strategies for Change. In *ICCHP '02*, Linz, Austria, July 15-20 2002, pp. 390-397.
- [Section508] Section 508 standards. <http://www.section508.gov>
- [Vaughan 03] Vaughan M. W., Degen H., Resnick M., Gremett P. (2003). Workshops: Best practices and future visions for search user interfaces: a workshop. CHI '03 extended abstracts on Human factors in computing systems, April 5–10, 2003, Ft. Lauderdale, Florida, USA, pp. 1052 – 1053

[Vegas03] Jesús Vegas, Pablo de la Fuente, Fabio Crestani: WebDocBall: A Graphical Visualization Tool for Web Search Results. ECIR 2003. Springer LNCS 2633, pp. 351-362

[W3C] W3C. Web Accessibility Initiative. <http://www.w3.org/wai>