Improving Search Engine Interfaces for Blind Users: a case study

Patrizia Andronico, Marina Buzzi

CNR-IIT via Moruzzi 1, 56124 Pisa, Italy

E-mail: Patrizia.Andronico@iit.cnr.it; Marina.Buzzi@iit.cnr.it

Carlos Castillo

Center for Web Research, University of Chile

E-mail: CCastillo@dcc.uchile.cl

Barbara Leporini(✉)

CNR-ISTI via Moruzzi 1, 56124 Pisa, Italy

E-mail: Barbara.Leporini@isti.cnr.it

Abstract: This article describes a research project aimed at improving search engine usability for sightless persons who use assistive technology to navigate the web. At the beginning of this research, a preliminary study was performed concerning accessibility and usability of search tools, and eight guidelines were formulated for designing search engine user interfaces. Then, the derived guidelines were applied in modifying the source code of Google's interface, while maintaining the same look and feel, in order to demonstrate that with very little effort it is possible to make interaction easier, more efficient, and less frustrating for sightless individuals. After providing a general overview of the project, the paper focuses on interface design and implementation.

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

Introduction

Due to the enormous amount of information available on the Internet today, search engines have become indispensable for finding specific information. This study describes 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 and drawbacks and propose improvements.

From the user's point of view, two main search engine components are equally important in order to perform 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 a search tool to set up a search task and explore the results, it is essential for user interfaces to be easy to use and accessible to all. This is particularly important for sightless users who interact via screen readers, as they perceive the page contents very differently and experience a much longer search time. This paper focuses on the needs of totally blind persons navigating via screen reader with a voice synthesizer, and no other assistive technology is taken into account. However, part of the obtained research results is also applicable in the case of visually impaired users.

The conducted investigation consists of four steps:

i. **Identifying the main search tool accessibility issues**

Automatic heuristic checking of accessibility and human control. The most popular search tools in Italy were selected, as well as others providing interesting features either in the search interface or in the presentation of results, as discussed in the following. Using two automatic evaluators, all the search tool interfaces were checked to detect the most important technical problems, according to W3C standards.

ii. **Investigating usability issues**

Empirical exploratory research and data analysis. A questionnaire was prepared in order to learn how individuals use search tools, their degree of satisfaction and their problems. An analysis of collected data was then performed [3].

iii. **Proposing specific guidelines**

Understanding interaction via screen reader. Google was chosen for the conducted study, since it is the most popular search engine used in the Italian sightless community. In this stage, problems of access via screen reader were highlighted by building a scenario of use for a simple search task with Google.

Definition of specific design guidelines. Based on the results of the conducted analysis, eight general principles that may improve the UIs interaction were proposed [12].

iv. **Applying proposed guidelines to Google**

Improving Google interfaces. At this point, Google interfaces were modified following the guidelines previously defined for adapting it to the needs of blind users. The original graphic style of the interface (i.e., the look and feel) was not changed. Only the source code was modified by

reestructuring it by adding some particular new features and using the CSS language, explained in detail later on in this paper. Additionally, two scenarios of use for a simple search task with the "original" and the "modified" Google Interfaces were also elaborated [1].

Introducing the screen reader and related problems when experiencing a page

A screen reader is an assistive technology used by visually impaired people to interact with the computer and other technological devices, such as mobile phones. Thus, a screen reader is a software which mediates between the user and the operating system (including its applications), assisting the visually impaired by interpreting the user interface. A screen reader can read aloud the content displayed by means of a voice synthesizer, or it can provide the same content in a written format by using a Braille display. In brief, when a user interacts via a screen reader, the page content must be interpreted and then converted into spoken or Braille output. That is, the screen reader recites in voice synthesizer modality the content of a page, word by word, line by line, etc., or it can write in a tangible Braille modality, so that the users can read it directly by themselves. Braille modality is mainly adopted and preferred by blind persons who learned the language at an early age and used it regularly over the years for studying, reading and so on. Most users are elderly people who, like elderly sighted people, are more reluctant to learn a new technology. Anyway, Braille output is even slower when working with a computer and with the Internet in particular. With the development of new technologies, the screen reader with voice synthesizer has become the most appreciated tool for blind people. On the other hand, it must be pointed out that using a screen reader is not so easy and, in the beginning, it requires particular effort for the user to learn. In fact, even blind people who are skilled in this technology may not know all the advanced features implemented by this assistive technology. It is also important to emphasize that page content perceived in vocal or Braille format is quite different from that perceived in visual modality. Several screen readers have been developed, such as Jaws for Windows [5], Windows Bridge (http://www.synthavoice.on.ca/), Hal for Windows (http://www.dolphincomputeraccess.com/products/hal.htm) and Windows eyes (http://www.gwmicro.com/). Jaws for Windows is the screen reader most often used and preferred by blind users in Italy.

In this paper, the term "screen reader" is used to indicate a screen reader with voice synthesizer. The screen reader announces every word on a page, line by line, sequentially. Concerning web pages, the new generation of screen readers interprets the HTML code as it is structured, thus considering different tags, while the first generation could not, generating a continuous line of information without any kind of separation. Despite the possibility of recognizing different tags in a page (such as tables, headings, lists, etc) some common problems are still present, such as the following:

- *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 and arrow keys, that are slower.

- *Excessive sequencing in reading the information* – Basic screen reader commands for navigation and reading only enable the user to explore page contents just sequentially.

- *Screen reader interpretation* – The screen reader deals with web page content in a very different way from 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 by the end user.

These drawbacks slow down navigation, annoying the user and often provoking great frustration. This is one of the reasons why some blind people do not like interacting with the Internet. In addition, it is necessary to take into consideration other user interface features that could affect web page accessibility and 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 achieve 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 code of a page.

- **Expressive power.** A visual element communicates any kind of information much more rapidly and effectively than other media. 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. In the case study presented in this paper, by means of shortcuts and "priority values" blind users can quickly reach the desired search field or result link. Moreover, assigning a simple and familiar label to a field could facilitate 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 supporting easy navigation. 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 memorize less clear components. This drawback can be observed in web directories whose interfaces are quite complex (full of elements), and can create confusion in a sighted user as well. The clarity of a page should be increased by using the CSS language for structuring the page in more logical sections for the user.

- **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 these powerful options are rarely used, even by skilled individuals. Also the function for searching within results (specifying additional keywords) could be easily used by unskilled users as well.

- **Result aggregation.** Some search tools offer information clustering, which permits users to explore results grouped by category and navigate a single branch of tree results at a time. If correctly implemented, this feature can increase interface usability and save time finding results.

Although search engine companies have frequently attempted to improve usability, the design layout is usually aimed at the sighted; thus in most cases new and interesting UI features are useless for blind persons. As a consequence, 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 requirements of users with diverse characteristics and ability. The increase of wireless technology, as well as the number of elderly persons in western societies, indicate that particular problems could be experienced by anyone in the near future. For example, a voice synthesizer installed in a car computer could be used to navigate the Internet while driving. Accessibility issues are becoming fundamental for all users regardless of disabilities when different contexts of use and different environments are taken into account in which people need to interact with diverse technological devices (i.e., smart phone, driving, walking, and so on).

Related Work

In the last few decades search technologies have evolved from dealing with small collections of homogeneous and structured data, to indexing the enormous amount of heterogeneous and unstructured data available on the Internet today. Before that, 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). Today, anyone can look up anything on the web, regardless of individual training or expertise in information retrieval. There are many differences between search tools and it mainly depends on the purpose of the tool itself. Within the framework of the Mann Library Project, Cornell University provides an interesting tutorial on "The Principle of Web Searching", which emphasizes the different kinds of indexing information, and differences between databases, but also provides guidance on how to evaluate the content found.

The rapid and successful spread of the Internet led to the development of a new kind of search tool: commercial search engines, such as Google, which provide free text searching of mostly unstructured data [21]. 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 [10] use search engines and search services to find information. However, current design solutions for search tool user interfaces have not yet achieved optimal usability, and even skilled individuals can encounter difficulties

when making queries or seeking information. [21] 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 60% reported finding relevant information most of the time. But this involves other problems, not only usability and accessibility of a search tool. As explained in [16], in fact, the way humans normally perform a search in long-term memory (LTM) can be divided into: *convergent searching*, when we some hints are available to help retrieve information ("comparing by similarity") and *divergent searching*, when there is not enough information or the available information is related to a generic domain ("frequency hazard"). Often humans naturally try one of these ways when looking for information with a search tool, often not coming up to any relevant result[1].

Adding the difficulties of web navigation to the complexity of the search engine's interface and functions, and to the difficulties in defining the right query, it becomes clear that it is particularly difficult for a blind person to use a search engine. Specifically, for people using a screen reader (which gives modal access) actions take longer and tasks are more difficult since additional actions are required [2]. The gap between blind and sighted users' efficiency when performing online search tasks is explored in [8]. In the related experiments, it was 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 (such as [17], [15]) but to the authors' 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.

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 [13]. In [22], a graphical visualization tool for helping users determine the

---

[1] There is abundance of literature regarding attention, user mental models, cognition and other topics of cognitive psychology, that explain the way humans use short-term and long-term memory for retrieving old and memorizing new information. It is useful to make a comparison between the human model and the way people look for information on the Internet, but this is not the scope of the present work. In this sense, it would be necessary to take into consideration other disabilities that can present different problems (i.e., deafness) as well as other groups of human beings (i.e., elderly, children, etc.).

relevance of a web page with respect to its structure is presented. Such tools can help the sighted user to 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 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 drafting a 2.0 version available at the URL http://www.w3.org/TR/WCAG20/.. In addition, other accessibility guidelines have been defined, i.e., the Section 508 standards defined by the US Government (http://www.section508.gov). An analysis of web content accessibility in the US private sector ([14]) has shown that how little attention is dedicated to this theme. In the reported study, 1080 organizations from different categories were contacted. Only 453 of them participated in the study and answered a questionnaire on web site accessibility. The study focused on the reasons for the inaccessible web sites (only 2% of the 1080 analyzed were accessible) and on understanding how to better achieve compliance to accessibility guidelines.

In January 2004, the Italian government approved a law[2] concerning "Provisions to support the access to information technologies for the disabled". A "study on the guidelines containing technical requirements for different accessibility levels and technical methodologies for verifying website accessibility"[3] was subsequently published by an Interministerial Committee.

To facilitate testing certain aspects of web content accessibility, several automatic tools, called validators, have been implemented. Validators can automatically check the code of a web page in conformity to the guidelines proposed by Section 508 as well as WCAG1.0. This is an initial and important step, but a human control on accessibility is still needed for those characteristics which cannot be

---

[2] Law 4 of 9/1/2004: Provisions to support the access to information technologies for the disabled, http://www.pubbliaccesso.gov.it/normative/law_20040109_n4.htm

[3] This study drafted by the "Methodology" and "Technical Rules" working groups of the Scientific Secretariat of the Standing Interministerial Committee is available at http://www.pubbliaccesso.gov.it/biblioteca/documentazione/guidelines_study/index.htm

controlled by a machine (i.e., link names, images used to convey information, etc).

Accessibility and usability requirements for sightless persons are discussed in [11], since both aspects are crucial for navigation via assistive devices. Another study concerning accessibility of on-line library resources for the sightless was performed in [18].

An interesting research focusing on search engine design for the blind is presented in [6], which describes and discusses the implementation of an auditory search engine prototype providing 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), 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. The interesting results are reported in details in [4]. In [8], a study was conducted to analyze the decision-making behavior and performance of blind and sighted users during the search task. This study 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.

Identifying the main accessibility issues

The investigation concerning usability and accessibility of search tools started with an automatic validation test of the following selected engines and directories, in order to verify whether their interfaces conformed to the W3C accessibility guidelines:

- **Google** (http://www.google.com/), and **Altavista** (http://www.altavista.com/), two very popular search engines;
- **Yahoo** (http://www.yahoo.com/) and **Excite** (http://www.excite.com/), which are web directories and meta-searches. Yahoo also has its own search engine.
- **HotBot,** a meta-search which permits customizing the user interface (http://www.hotbot.com/);

- **Vivisimo,** a meta-search which performs on-fly clustering of results (http://vivisimo.com/);

- **Kartoo,** a meta-search which displays results through a series of interactive maps (http://www.kartoo.net/).

For cross-testing, two free validators were chosen: Bobby[4] (http://bobby.cast.org/) and Torquemada[5] (http://www.webxtutti.it/). It was observed that Bobby conveniently arranges errors by priority level as defined in WCAG 1.0, while Torquemada lists errors in the same order in which they appear in the source code. Furthermore, Bobby's report is more complete than that of Torquemada, which is still being developed. The tests were performed on four types of interfaces[6] 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), the validator output was manually checked.

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 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.

---

[4] "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

[5] "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

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

Investigating usability issues

Since a sighted user may also have difficulties using search tools, testing with different categories of users (blind and sighted) is important in order to discover the source of each problem and compare results.

To gather user feedback, a four-part questionnaire was prepared, comprising user characterization, general knowledge of search tools, use of search interfaces for simple and advanced search, and interfaces for setting preferences. The information was collected and used in absolute confidentiality.

The questionnaire was distributed to individuals living in Italy. 52 answers were received, but the majority of these were from sighted users; the total sample comprised 75% sighted and 25% sightless users; 33% were women and 67% men; age range was from 20 to 60+ years. Of the respondents, 54% were ICT-skilled; all subjects use the computer either at home (13%), work (44%) or both (43%).

Since the sample was actually unbalanced (25% of blind and 75% of sighted users), data were normalized for each single user category in order to compare results. The graphs in Fig. 1 and Fig. 2 show the response of blind individuals in yellow solid color (percentage calculated only on blind users) and those of the sighted in a blue diagonal-row pattern (percentage elaborated only on sighted users).

Of all the search tools analyzed, Google was the most popular, followed by Altavista and Yahoo (Fig. 1a). A first observation was that sightless users do not use web directories. In fact, for the blind, an interface crowded with elements is very difficult due to the more complex visual layout and structure, and it may become impossible to use. It also emerged that 44% of sighted users utilize different search tools, in contrast with habits of blind users who always (56%) and almost always (23%) utilize the same tool, i.e., the same interface (Fig. 1b).

Feedback on knowledge of search tools showed that 62% of blind users tried to use different tools, vs 85 % of sighted users (Fig. 2a). In addition, only 23 % of blind users, vs 70% of sighted users, perform search frequently (Fig. 2b), highlighting the difficulty of interaction via a screen reader, as well as the importance and the impact of simplifying interaction.
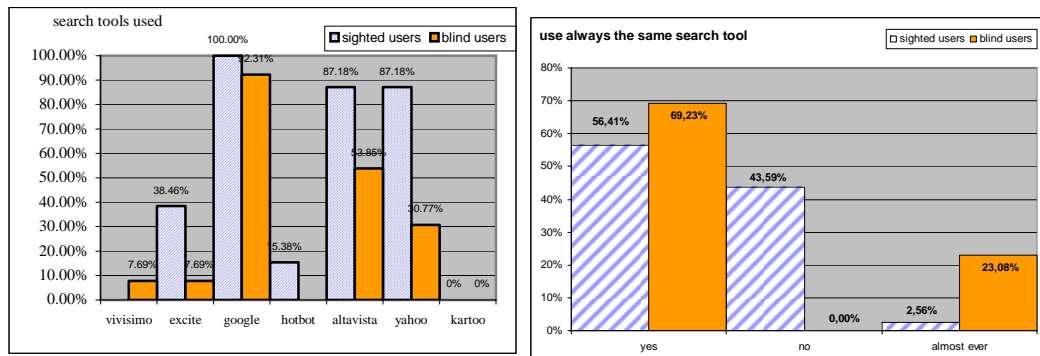
Fig. 1. - **a)** Search tools used; **b)** Always use the same search tool
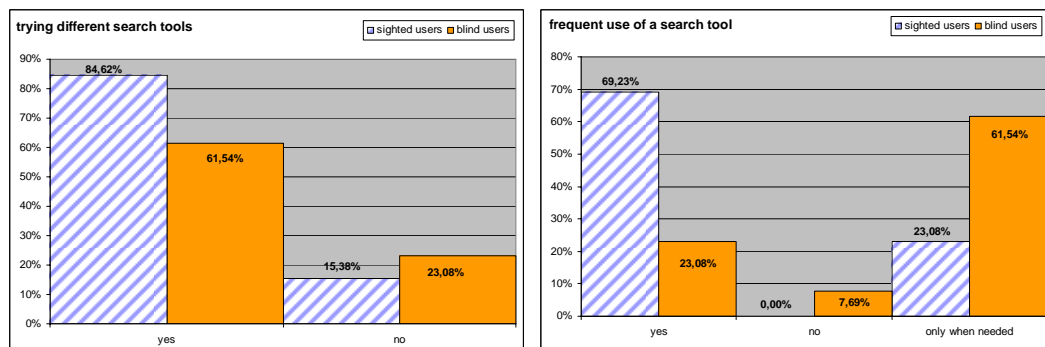

Fig. 2. - a) Use of different search tools; **b**) Frequency of search tool use

The third part of the questionnaire concerned how users utilize search engines. Only 28% of the sighted and 15% of the blind users had attempted to configure the search tool (i.e., the preference page). Furthermore, only 38% of blinds (compared to 87% of sighted users) have used the advanced search. This data confirms that interaction with a more complex interface is more difficult for the blind. However, both blind (85%) and sighted (87%) agree that using a search engine is the fastest way to find information on the Internet.

Last, 92% of sighted users think that search engines are easy to use, while 77% of blind users specify "not always". 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 3a. 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. 3b).
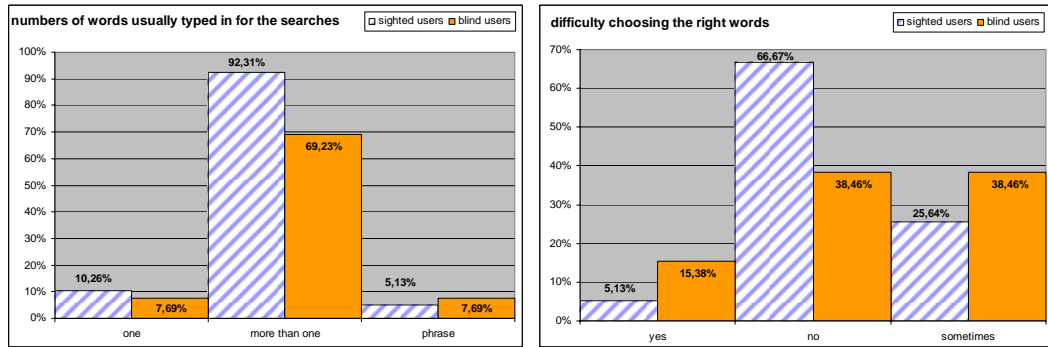
Fig. 3. - **a)** Number of words used in query; **b)** Difficulty in choosing the right words

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 results (Fig. 4a). 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 sequential access to page contents. Thus, the blind need more time to visit each result page and accessing more than two pages becomes even more difficult, if options for rapid navigation via keyboard are not offered.
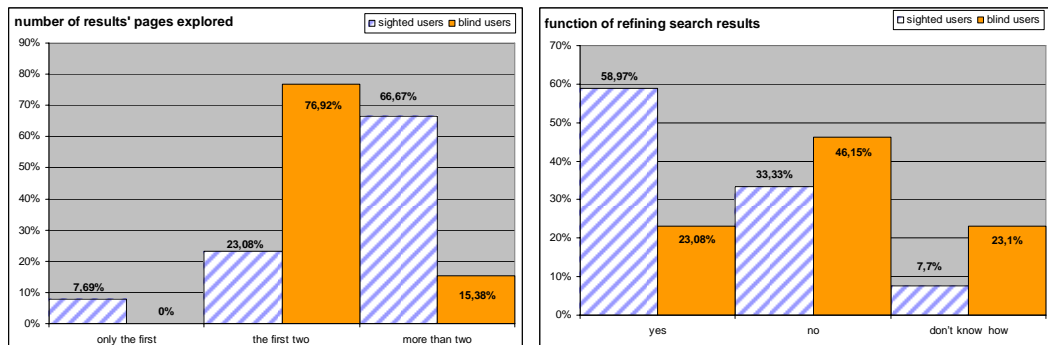
Fig. 4. - **a)** Page of results visited;   **b)** Refining search results

In addition, only 23% of blind users use the refining function (for searching into results), compared to 59% of sighted users (Fig. 4b). 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 5a 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. 5b).

Fig. 5. – **a)** Difficulties during search; **b)** Successful search

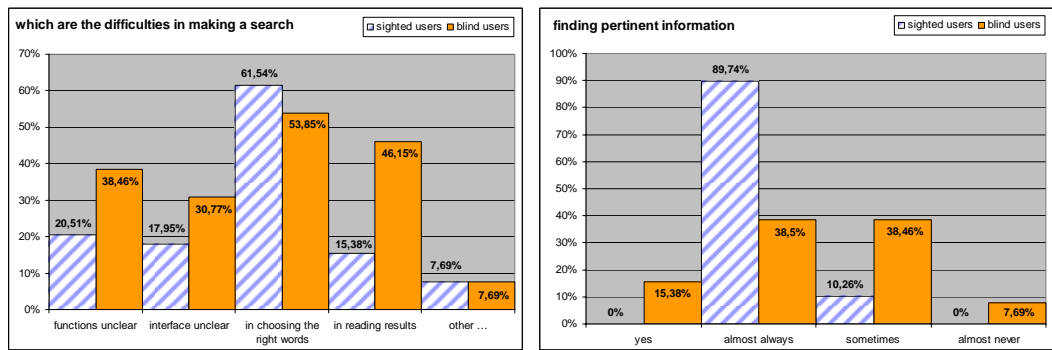These data show that sightless individuals are more sensitive to user interface design. Thus, it is crucial to structure results in a way that permits blind users to access them quickly and easily. For instance, reaching the result area rapidly, clearly knowing the number of results obtained, reading appropriate links about the results, etc., are important features for the blind user.

Proposed guidelines

The main philosophy of User-Centered Design methodology (UCD) is that the individual is the center of any kind of artifact created to satisfy human needs. It can be assumed that a user interface is a kind of artifact, and should be designed with the users, keeping in mind their particular experience and cultural environment. It is important for the designer and the developer to take all these issues into account from the very early phases of a design project.

For users interacting with any kind of assistive technologies, the UI layout and structure are crucial. When navigating via 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 users perceive page content and interact with the interface. The main related issues are:

A. **Page content serialization.** The JAWS screen reader takes the page source and serializes its content (link, edit field, button, cell, etc). 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 coded 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

14

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 links, applying shortcuts to main elements, using specific tags such as <Hn>, etc. 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, etc. 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 other 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, the following principles are proposed to be considered when designing a search engine interface layout:

1. **Easy location and labeling of edit field and search options.** Place edit fields, option buttons and any other search element at the top of the web page; avoid secondary elements (links, texts, banner frames, etc.). To place an object in a specific position of the visual layout, use the position CSS properties. Take care of correctly matching <label for> with input elements, and placing 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 on 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.** Place the list of the result links with their summaries immediately following the search result notification (nothing else should be located in the middle). Number the results; for instance the <ol> or <ul> tags may be used to this purpose. This feature enables a screen reader to inform the

user of the number of items; the user is then able to skip quickly, item by item. As a default, the page should contain a maximum of ten items.

4. **Recognizing sponsored links.** Keep sponsored links separate from the other results. Use the CSS positioning property to locate sponsored links on the right side of the page, or if a table is used put a clear label in the summary attribute, and insert the table code after the results list on the page source.

5. **Adding navigation and help links.** Place 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 (using the tab index attribute) so users can reach the most important elements quickly. On the home page (i.e., simple search) higher values should be assigned to edit field and search options; on the result page, 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.

Applying proposed guidelines to Google

Following the elaboration of the guidelines, it was decided to apply them to a specific case. The original Google interface (source code) was modified for simple search and result exploration, while carefully maintaining the original graphic layout. In the following, these new interfaces are referred to as "modified

Google User Interfaces". The new interface design was intended to map the visual functions to aural-usable information, in order to achieve rapid positioning in the desired interface area, immediate information on search status, etc. Since the aim was to improve the user interface 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.

The following section of the paper describes the performed reengineering in technical details.

Reengineering of Google UIs

The main idea behind this part of the work is to re-engineer the Google code by applying the aforementioned guidelines to web search pages facilitating interaction by screen reader, in order to shorten the time needed to carry out a search task.

Reengineering the user interfaces required a complete redesign process, but without modifying the visual layout, i.e., maintaining the original look and feel. To accomplish this, the page's information was separated from the visual rendering, using the cascading style sheet (CSS), one of the W3C Recommendations, which 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, the first performed steps were an attempt to re-write the code in accordance with the most recent standard on the web:

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

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

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

For this study, Internet Explorer was used, since the Jaws screen reader is tested with this browser and does not work properly with other ones.

Google offers the following slightly different versions:

- the original one (.com)
- "localized" versions (such as .it, .uk and .fr). This versions provide options (radio buttons) to restrict the search to the specific country domain (it, .uk, .fr.) or language (if different from English).

Since English was needed <mark>for communicating our results</mark> to the international community and Italian was needed for conducting the usability test, it was decided to work in parallel with the Italian (http://www.google.it/) and the UK (http://www.google.co.uk/) versions. It was in fact chosen to modify http://www.google.co.uk/ and http://www.google.co.it/, i.e. the "local" versions of Google for two main reasons. The first one is due to the fact that Google's local versions are commonly utilized by people, because of the automatic redirection of the original .com onto the user's country version. The second reason is strictly related to the technical differences between the local and the .com version, resulting the former more complicated than the latter one. Local home pages, in fact, provide additional options (radio buttons) to restrict the search to the specific country domain (.uk, .it, etc.) or language (if different from English), while the result pages, for not native-English regions, presents, for each result, an additional link, such as "translate this page".

For testing the implementation, Jaws for Windows v. 5.10 and the browser IE v. 5.5 e v. 6.0 were used, since Jaws does not work properly with other www clients, such as Netscape Navigator or Mozilla Firefox. Since JAWS is a fairly complex program itself, requiring considerable knowledge to be used with maximum proficiency, variations in the JAWS user's knowledge can contribute substantially to variations in web surfing abilities in general, and in search engine usage in particular. Thus, in the current implementation, in order for the new interface to be accessible for every user, only the most common JAWS commands were used, excluding the more advanced commands.

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.

Additionally, Google uses tables just for layout purposes in both the simple and advanced search UIs. For this reason, the first aim of the implementation was to demonstrate that with very little effort it is possible, in an interface as simple as Google, to eliminate the tables maintaining the original layout.

Re-write the code according to the standards

Fig. 6 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, however, their interpretation by Jaws is very different (as shown in Fig. 11).
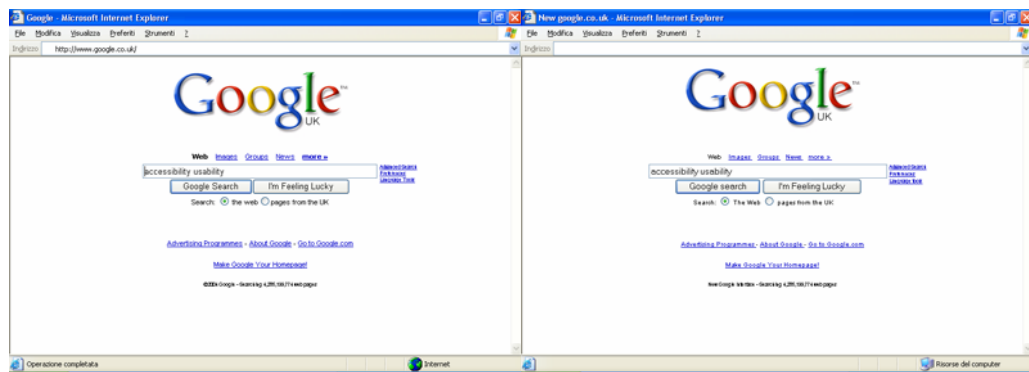
Fig. 6 – a) The Google home page available at http://www.google.co.uk b) The modified interface

Looking more closely at the source code of the original Google UI, it can be observed that two tables exist, which are invisible on the screen since the cell border is set to 0 pixels. Fig. 7 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.

Fig. 7 – Use of tables in the UK Google home page

In the reimplementation, it was decided to follow the same logical structure of the original graphical layout, dividing the interface into four sections (highlighted in red in Figure 8):

  1) Navigation bar

2) Search box and options

3) Advanced Search and Preferences

4) Google info and other links (Google Links)
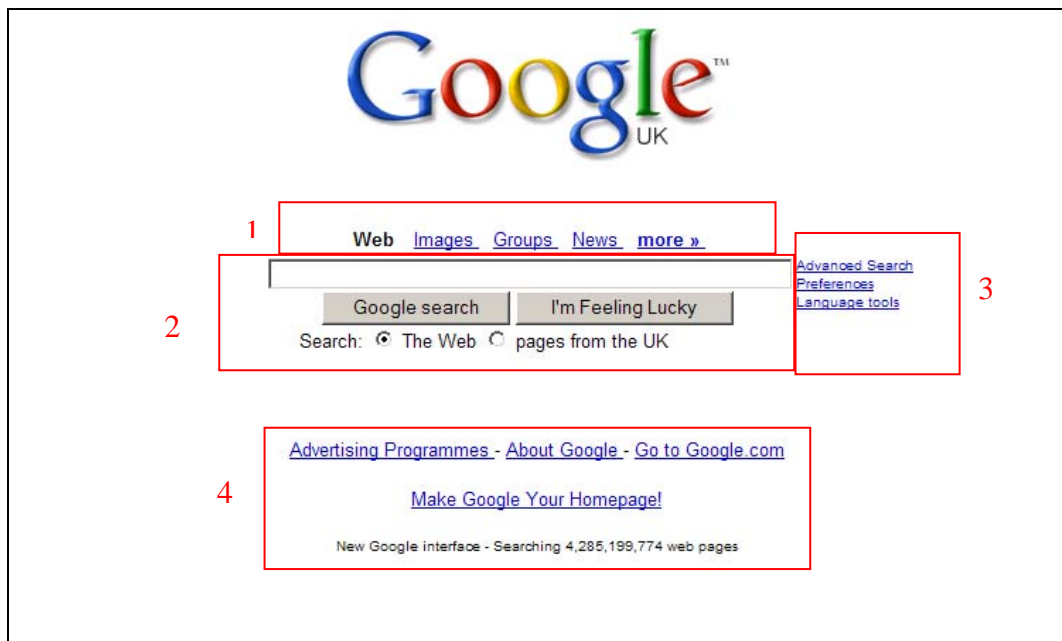
Fig. 8 – Logical sections of the modified Google home page

In order to define those four sections, the CSS block properties were used rather instead of the layout tables. In practice, <DIV> tags were used 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. The goal is to provide a similar opportunity to users navigating by screen reader. Since content structuring was based 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 the 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 needs to be 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, through the "title" attribute list, can quickly learn the page structure and easily jump between sections.

The same technique was used for the results page, dividing it into eight sections as shown in Fig. 9.
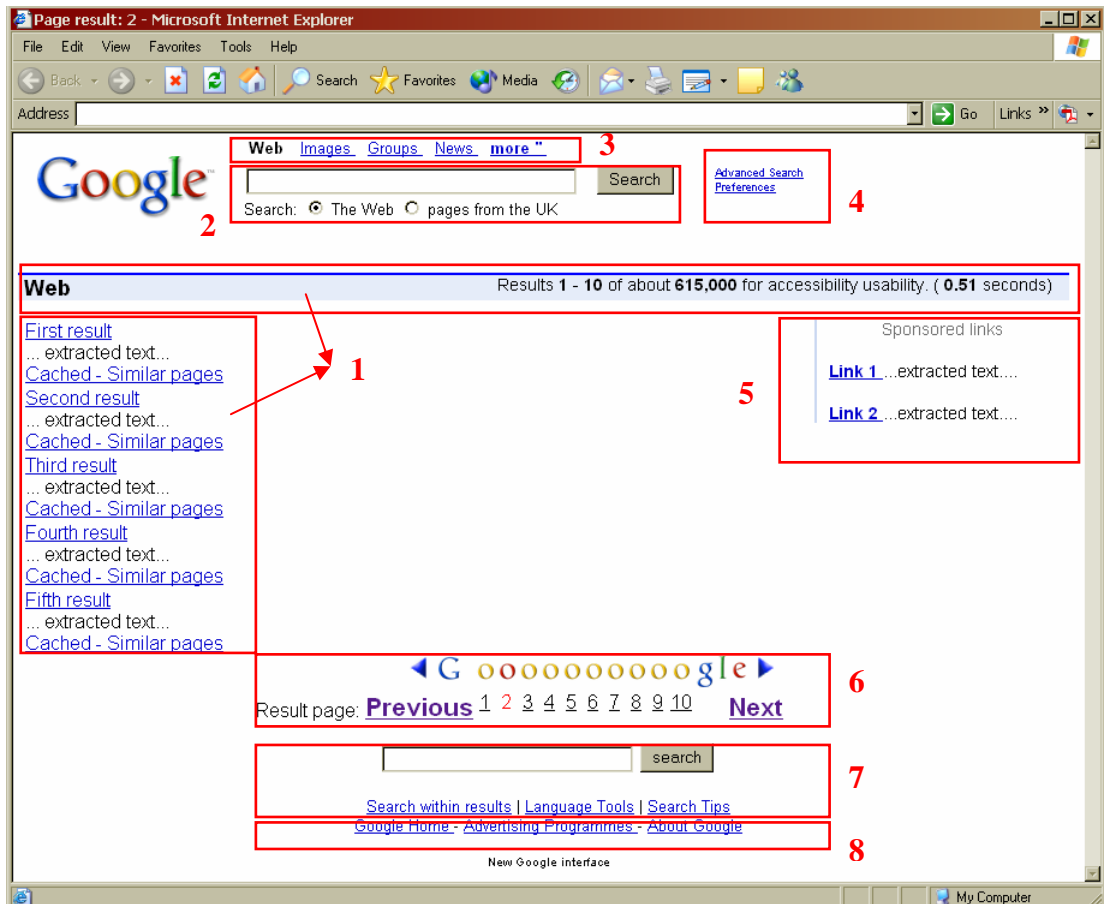
Fig. 9 – Modified Google interface: results page

One of the features provided by Jaws is the possibility of recognizing the heading levels - i.e., content enclosed between <Hn> heading tag 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 a good document structure (e.g., chapters, paragraphs, and so on), they can be utilised for structuring the web interface. The idea is to assign a heading level <Hi> to the "title" (in the specific case a hidden label) of each page section. Practically, this approach "extends" the heading level usage. Heading levels can be applied for obtaining a good page structure for documents and for 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. Using this method, an "index" of the sections available for the page can be created "on the fly" (i.e., by a special Jaws command "Insert+F6"), thus making substantially easier to move to a desired section or to navigate among the headings (previous and next).

Fig. 10 shows the Heading Lists of the Google modified interfaces (home page and result page), generated by Jaws. Four headings were defined 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".
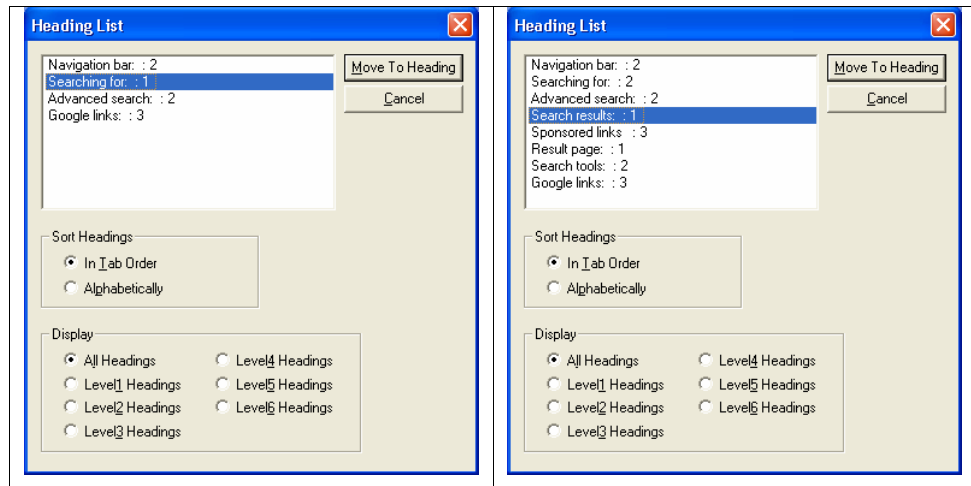
Fig. 10 – Logical sections of the modified UIs: a) home page and b) results page (generated by a specific Jaws command)

Thanks to this structure, blind users can jump to a specific part of the interface using three different Jaws commands:

o By pressing the Insert+F6 key combination, the user obtains the heading list. With the arrow keys he/she can select the desired section and just push the Enter key.

o 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.

o By pressing the "i" key ($1 <= i <= 6$), the user jumps to the next section associated at the Heading Level <hi>. By pressing "shift+i" ($1 <= i <= 6$) the focus moves to the previous section associate to a tag <hi>. 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. Pressing once again "1", the focus moves to the next section associated to a <h1> tag, i.e. the Result page. In any case, other specific commands for skipping to the first or last heading are available.

New logical order

Once it was decided which sections to consider on the page, the CSS file was defined 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 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 GUI[7] is not necessarily the first element in the XHTML code.

The order of the blocks is quite important, both for a correct interpretation of the contents, 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, it was 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, it was not possible to use the absolute positioning property of the CSS, since some browsers, such as Internet Explorer v. 6 (the browser used in the current investigation) still do not support this feature. For this reason, a compromise was achieved between the best technical solution and one that is actually feasible, due to the current state of screen reader and browser compliance to standards.

Fig. 11 reports the source code in both the original (a) and the modified (b) Google home page. Although the source code is increased in (b) with respect to (a), a blind user can easily jump to the desired section, or navigate sequentially after visiting the most important sections first.

---

[7] It is assume that the first element on the screen is the first element encountered by the user's eyes when reading or scannig a page (according to European culture, left-right and top-bottom).

```<table cellSpacing=0 cellPadding=4 border=0>```	```<!-- Navigation bar -->```
```<tbody> <tr>```	```<div id="navigationbar">```
```<td class=q noWrap>```	```<h2 class="hidden-label"> Navigation bar: </h2>```
``` Web ```	```<!-- xhtml code for the navigation links -->```
```<!-- code of all the navigation bar links -->```	```</div>```
```</td> </tr> </tbody>```	
```</table>```	```<!-- search area -->```
```<table cellSpacing=0 cellPadding=0>```	```<div id="left">```
```<tbody> <TR>```	```<div id="sfields">```
```<td width="25%"> </TD>```	```<h1 class="hidden-label"> Searching for: </h1>```
```<td align=middle>```	```<!-- search edit field and buttons -->```
```<!-- search edit field and buttons -->```	```</div> <!-- End of sfields block -->```
```</td>```	```<div id="schoices">```
```<td vAlign=top noWrap width="25%">```	```<!-- code for the search radio buttons -->```
```<!-- links for advanced search -->```	```</div> <!-- end of schoice block -->```
```</td> </tr>```	```</div> <!-- end of left block -->```
```<tr>```	
```<td align=middle colSpan=3>```	```<!— Advanced Search and Preferences links -->```
```<!-- search radio buttons -->```	```<div id="right">```
```</td> </tr> </tbody>```	```<h2 class="hidden-label"> Advanced Search: </h2>```
```</table>```	``````
```<p>```	```Advanced Search ```
``````	```...```
```Advertising Programmes </A> -```	```</div> <!-- end of right block -->```
```...```	```<div id="bottom">```
```</p>```	```<h2 class="hidden-label"> Google Links: </h2>```
	``````
	```Advertising Programmes -```
	```...```
	```</div> <!- end of bottom block -->```

Fig. 11 – Source code showing the sequence of content blocks: a) the original, and b) the modified UK Google home page

Fig. 12 shows the screen reader's interpretation of the original and of the modified page, associated with the order of the section in the corresponding graphic interfaces. Italics refer to words/sentences read aloud by the screen reader, informing the user about interface elements (link, button, edit field, heading level, etc). New parts, added when reengineering the interface, are highlighted in bold.

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

Fig. 12 – Screen reader interpretation of the: a) original, and b) the modified UK Google home page

Additional features

Once the page was restyled, it was decided to improve accessibility and usability characteristics by including some additional features, explained in detail in the following paragraphs.

Quick navigation

In order to improve navigation among the elements in the sections by keyboard, it is advisable to use tabbing order and shortcuts. For this reason, some tab-index and access keys were assigned 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 the user. For example, we chose the letter H for navigation help, or G for the Google home page. Table 1 summarizes all the shortcuts introduced in the modified Google UI.

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

Different levels of tabbing order were applied for the home page and the results interface. In the home page, the search field and options are more relevant than other parts; in the results page, at 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 on those pages. The next table summarizes the visiting order when the user moves by Tab key on both the Google home page and the results page.

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

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

Fig. 13 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 <a name="results"… >. This tag is defined to create a false link, which allows users who press 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"> <img height=26 alt="" src="img/nav_previous.gif"
width=68 border=0> <br />
<strong>Previous</strong></a> </div>
...
<div class="nav-pages">
<a href="" accesskey="+" tabindex="5"> <img height=26 alt="" src="img/nav_next.gif"
width=100 border=0> <br />
<strong>Next</strong></a> </div>
```

Fig. 13 - Fragment of result page code containing tab index and access key assignments

Hidden labels

"Hidden labels" are another feature which has been added in order to simplify navigation for blind users. In this case, a label added to a particular section is hidden from sighted users, 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, hidden labels aim to facilitate recognition of an element in a page, whereas no attribute does this.

Hidden labels are used the 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 a 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. Such sounds are called "sound icons", since each aural message has a specific meaning.

This feature was applied to the Google interfaces by 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 blind users, because it immediately informs them 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 recognize and interpret aural css properties, the developed interface uses javascript, as shown in Fig. 14.

```
//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];
}
```

Fig. 14 - 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 static, simulated pages could have been used for the search engine's results, in order to carry out user testing the modified Google interface with "live" results was implemented. In order to do this, it was necessary 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 the 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 the developed application), SMTP (SOAP over E-mail), and 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 15. 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 15 - Extract of the response to a query from the Google APIs. Details such as data types and name spaces are omitted for clarity's sake.

The results are structured in a simple way. Each single result of the query (<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, the XSLT (eXtensible Stylesheet Language Transformations) language is used, which describes the transformation of a source XML document into a destination XML document. In the specific 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">
     <li>
     <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>
     </li>
</xsl:template>
```

Figure 16 - 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 XLST transformations can be provided by the web server and executed by the browser, providing different renderings for different users. However, in the current case study, it was important to provide a single interface for all users, with the same layout but with hidden markup for improving accessibility and usability.

Unfortunately, the index used by Google APIs is different (smaller) from those used by Google.com and regional versions, since it is intended for research use only. Thus, the same query do not produce the same results when the user sends a query using the browser and via APIs. For this reason, in order to set up the environment for user testing, it was necessary to create two transformations, one re-creating the original Google Interface and the other for the modified Google Interface.

Limitations

It is important to notice that different blind computer users may use different versions of the various screen reader programs, and these variations can cause different outcomes, thus producing a great difference in accessibility.

Concerning implementation, due to the fact that Jaws is only available for Windows, the developed interface runs well with IE but not with Netscape and Mozilla; however, basically all of the transformations we have shown can be trivialy made compatible with any standards-compliant browser. Given the recent changes in the market share of different web browsing software, we expect screen reader software to support non-microsoft browsers in the near future. Javascript was used to activate different sounds for communicating important events immediately to the user, such as "the focus is on the search box", or whether the search query produced results or not. Obviously, this additional feature is not accessible if the user uses a textual browser such as Lynx or a very old browser, or if Java script is disabled. When browsers and screen readers are able to interpret aural CSS properties, this type of feature should be available with the appropriate style sheet.

Conclusion and Further Work

The analysis presented in this paper aims to demonstrate that usability for blind users can be greatly improved while maintaining an appealing graphic layout. To

accomplish this, the Google user interfaces were redesigned by taking into account problems of navigation via screen reader.

First, the UI was structured in logical sections, grouping interface elements by function. Interaction via screen reader was then simplified. Specifically, heading levels were used to assign different degrees of relevance to different parts of the interface (i.e., logical sections), and these sections were marked to be perceived only by screen reader by using "hidden labels". 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, it is planned to carry out a specific user test with sightless users. During the test, several tasks will be assigned to participants in order to actually evaluate the effects of the proposed principles. The underlying objective is to demonstrate that using the redesigned interface instead of the original one a blind user can perform fewer steps to accomplish a specific search task.

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 great for any user.

The source code used for this research is available, for further information please contact the authors for licensing details.

References

[1]   Andronico P, Buzzi M, Leporini B (2005) Increasing Usability of Search Engine Interfaces for the Blind. To appear In: Proceedings of HCII 2005, USA

[2]   Brajnik G. (2004) Achieving Universal Web Access through Specialized User Interfaces. In: Adjunct Proc. of 8th ERCIM UI4ALL Workshop. Vienna, Austria

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

[4]   Craven J, Brophy P (2003) Non-visual access to the digital library: the use of digital library interfaces by blind and visually impaired people. In: Tech. Report, Manchester: Centre for Research in Library and Information Management. http://www.cerlim.ac.uk/projects/nova/nova_final_report.pdf

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

[6]   Ferworn A., Bodner R, Chignell M H (2000) Auditory WWW Search Tools. In: Proceedings of Int. Conference on Auditory Display 2002. Atlanta, USA, April 2-5, 2000. http://www.icad.org/websiteV2.0/Conferences/ICAD2000/ICAD2000.html

[7] Hearst M (1999) User Interfaces and Visualization. In Modern Information Retrieval by Ricardo Baeza-Yates and Berthier Ribeiro-Neto, by Addison Wesley, pp.257-323.

[8] Ivory M Y, Yu S, Gronemyer K (2004) Search result exploration: a preliminary study of blind and sighted users' decision making and performance. In: CHI 2004 Extended Abstracts, pp. 1453-1456.

[9] JAWS for Windows, Freedom Scientific. http://www.freedomscientific.com

[10] Kobayashi M, Takeda K (2000) Information retrieval on the Web. ACM Computing Surveys Vol. 32 (2000), pp. 144- 173

[11] Leporini B, Paternò F (2004) 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

[12] 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

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

[14] Milliman R E (2002) Website accessibility and the private sector: disability stakeholders cannot tolerate 2% access! http://www.rit.edu/~easi/itd/itdv08n2/milliman.htm

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

[16] Reason J (1990) *Human error*, Cambridge University Press

[17] Scapin D L and al (2000). In: Proceedings of the Conf. on Human Factors and the Web. Austin, Texas, 19 June 2000. http://www.tri.sbc.com/hfweb/scapin/Scapin.html

[18] Schmetzke A (2002) The Accessibility of Online Library Resources for People with Print Disabilities: Research and Strategies for Change. In: Proceedings of *ICCHP '02*, Linz, Austria, July 15-20 2002, pp. 390-397

[19] Section 508 standards. http://www.section508.gov

[20] Turner T, Davis P, Morris-Knower J (2002) Project at the Library of the Cornell University, Tutorial on "The Principle of the Web Searching", http://www.mannlib.cornell.edu/reference/tutorials/search/ Cited 2005

[21] Vaughan M W, Degen H, Resnick M, Gremett P (2003) Workshops: Best practices and future visions for search user interfaces: a workshop. In: Proceedings of CHI '03 extended abstracts on Human factors in computing systems, April 5–10, 2003, Florida, USA, pp. 1052 – 1053

[22] Vegas J, de la Fuente P, Crestani F (2003) WebDocBall: A Graphical Visualization Tool for Web Search Results. In: Proceedings of ECIR 2003. Springer LNCS 2633, pp. 351-362

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