Long paper

Increasing usability when interacting through screen readers

Barbara Leporini (✉) · Fabio Paternò

B. Leporini · F. Paternò
ISTI-CNR, Pisa, Italy

B. Leporini
Department of Computer Science, University of Pisa, Pisa, Italy

✉   B. Leporini
E-mail: barbara.leporini@isti.cnr.it

Abstract The application of appropriate Web site design and evaluation methods helps to ensure more usable and accessible Web sites. While in the literature guidelines and evaluation methods for accessibility and usability are given and discussed separately, we aim at identifying the relationships between these two concepts, in particular considering usability criteria for accessible Web sites. In this work, we propose a set of such criteria targeted to improving the navigability for a specific group of disabled users, i.e., vision-impaired people. The identification of the eighteen criteria suggested herein was performed through empirical feedback, by which potential issues were identified. Subsequently, a systematic method was developed on the basis of the analysis of potential solutions, resulting in a classification of the criteria according to usability aspects. Some example applications of the proposed criteria to three existing public administration Web sites are discussed.

Keywords Please · provide · five · keywords · here

1 Introduction

In recent years, the use of Web sites has been widening, and the number of users accessing them is steadily increasing. For this reason, it is important that the information be easily reachable by all, including people with disabilities. The difficulties in providing universal access can be addressed through the application of the principles of usability and accessibility. A Web site can be said to be accessible if it can be used by everyone, including people with disabilities. Usability is a multidimensional concept, since it can refer to several aspects whose importance depends on the application domain. Although accessibility and usability are closely related, accessibility is aimed more specifically at making a Web site available to a wider user population, while the goal of usability is to make users' experience with the Web site more efficient and satisfying.

Usually usability and accessibility issues are treated separately. Our aim herein is to identify their relationships: in particular, we aim at addressing the meaning of usability in the context of Web sites accessible to disabled users. Often, when designers consider people with special needs, they tend to address only accessibility issues, and ignore those regarding usability. Technical accessibility is a pre-condition for usability. However, even if a site is in principle accessible, because it completely complies with technical accessibility standards, it can still be so hard to use for people with disabilities that they may not succeed in reaching their goals. In our work we consider, in particular, accessibility-usability issues for blind or vision-impaired people. Therefore, we consider the context in which these users work, i.e., Web browsers, particular devices (voice synthesizers or Braille displays), particular programs (screen readers, screen magnifiers) and so on. When interacting with a screen reader, the way in which the information is set out on the page is very important, because what the blind user hears is very different from what sighted users would read from the screen. Thus, blind and vision-impaired people were involved during the stage of identification of possible criteria, through preliminary tests on some selected examples. In this paper we present a refined version of the criteria that were introduced in [9], and provide examples of their application to three public administration Web sites.

We first discuss related work in this area to better frame our contribution. Next, we introduce the proposed criteria for improving the usability of accessible applications, and classify them according to the standard definition of usability. Then, we provide some examples resulting from application of the criteria. Lastly, we provide some concluding remarks and indications for further work.

2 Related work

Usability and accessibility are two concepts that apply not only to Web sites, but in general to all interactive systems. Several evaluation techniques have been proposed in the literature, some of which have been widely used. A review of evaluation methods with automatic support is available in [1], where a classification is proposed. In order to obtain more usable and accessible Web sites, the developer has to follow well-defined criteria and guidelines. Several usability guidelines have been formulated. Most accessibility issues are taken into account especially by W3C (the World Wide Web Consortium) in the Web Accessibility Initiative (WAI) [8], that put forward a number of recommendations and guidelines to promote Web accessibility ("Web Content Accessibility Guidelines 1.0", WAI 1999). The guidelines are intended for all Web content developers (page authors and site designers) and for developers of authoring tools. Such guidelines mainly focus on Web accessibility aspects for people with disabilities, though, in our opinion, they do not address usability at a sufficient level in this context. Hence, we intend to investigate those aspects that act on the organization and rendering of information in the page. In practice, we would like to extend accessibility guidelines, while also aiming at improving and facilitating task performance. A Web static analyzer (WebSAT) is a tool that verifies, referring to several guidelines, the HTML code of Web pages, in order to point out potential usability problems [5]. It carries out a usability evaluation, but does not provide any suggestions to the developer. Some other tools, such as Bobby [2, 3], and LIFT [4], perform accessibility evaluations based on criteria and guidelines. Although such tools are capable of revealing accessibility problems, they exhibit several drawbacks, such as rather long reports that are difficult to understand for inexperienced developers, no automatic support for the repair of the identified problems, a limited number of potential accessibility problems that can be detected, etc. These tools reveal accessibility problems, regardless of usability aspects for persons with disabilities. In fact, although LIFT does partially support usability evaluation, usability for special users can not be sufficiently assessed (e.g., theconsistency of text and background colours are checked, but not specifically for vision-impaired people).

There are various international projects involving the accessibility and usability of user interfaces for people with special needs. Stephanidis' group has long been working on user interfaces for all, elaborating methods and tools allowing the development of unified user interfaces [6, 7]. Although guidelines and tools have been taken into account, they are mainly related to accessibility issues. In our approach, we want to consider not only accessibility, but also usability for accessible Web sites in the context of common Web browsers.

3 The proposed criteria for the usability of accessible Web sites

In this section, we discuss the proposed criteria to improve the usability of accessible Web sites. We suppose that a Web site is already accessible (i.e., it complies with accessibility guidelines), and we describe those aspects of Web pages that have an effect on navigability by users with disabilities, and more specifically blind or vision-impaired users). We propose a set of criteria that can be used to support both design and evaluation. We have focused on the Web page code, taking into account the HTML/XHTML language, JavaScript and style sheets (CSSs). Accessibility guidelines advise not to use JavaScript because some Web browsers may not support it. Even though we share such principles, we aim at providing possible suggestions that also involve JavaScript, because we do not wish to excessively limit the means at the disposal of developers. Moreover, this is especially important for sites already including scripts, which have to be modified in order to fix potential usability problems. Our goal is to create a semi-automatic environment supporting the designer rather than a completely automatic solution that would be too restrictive. In fact, developers may often decide not to repair Web sites because of the effort required, which depends on the number of changes necessary, and sometimes requires a general reorganization of the Web site.

In defining the proposed criteria, we aimed at identifying the main aspects that can cause usability problems in accessible Web sites. Then, for each criterion, we provide more technical solutions to reach that goal, taking in account developers' choices in building the Web site (e.g., frames or JavaScripts). Thus, usability aspects are referred to in terms of associated criteria, while the technical solutions in terms of checkpoints.

3.1 How the criteria is organized

We consider the aspects that can be potential usability problems and the associated checkpoints. A *checkpoint* is a specific construct in a language for Web page development (for example, a radio button, a frame, etc). While a certain criterion can involve only one checkpoint, others can address multiple checkpoints.

The classification of our criteria differs from that of the W3C accessibility guidelines, which are arranged in 3 layers: guideline statements (i.e., general principles of accessibility); a checkpoint list (i.e., how the guideline applies in typical content development scenarios); and a techniques

section (i.e., implementations and examples for the checkpoints). Our approach aims at providing developers and evaluators with a more compact organization of criteria in order to simplify their use.

Moreover, we group our proposed criteria according to the standard usability definition. More precisely, first of all we classify the proposed criteria according to effectiveness, efficiency and satisfaction principles; secondly, we sub-classify them depending on the type of impact on the user interface.

Classifying criteria according to the usability definition (ISO 9241) means that we identify those that are most important to reach the users' goals (effectiveness), those that allow reaching them more quickly (efficiency), and those that best satisfy users (satisfaction). We consider effectiveness criteria more important than those based on efficiency and satisfaction, because failure to satisfy such criteria could lead to users' not being able to accomplish their tasks. Thus, we consider effectiveness (level 1) more important than efficiency (level 2), which in its turn is considered more important than satisfaction (level 3).

The other dimension by which we classify criteria is the affected user interface aspect. A user interface is composed of two main components: the presentation, indicating how the user interface provides information to the user, and the dialogue, describing how the actions that users and system perform can be sequenced. According to this, we have designated "a" the presentation criteria and "b" the dialogue criteria.

3.2 The proposed accessibility usability criteria

In our work we determined 18 criteria to improve accessible Web site usability for users who access Web pages via a screen reader. As mentioned above, the criteria have been grouped into three sub-sets: effectiveness (six criteria), efficiency (nine criteria) and satisfaction (three criteria). For each criterion, several checkpoints are proposed in order to indicate how it can be applied. To identify the criteria, our study was structured in various phases:

– *Empirical identification of potential issues*. First of all, we took into account those aspects (see below) that can become potential navigational problems for users using a screen reader or magnifier. For this purpose, we collected feedback from a number of blind people (including one of the authors).

– *Analysis and testing of potential solutions*. According to the aspects identified in the previous phase, we analysed HTML specifications and JavaScripts, to determine possible solutions.

Then, based on our hypothesis on possible criteria to solve the problems identified, some simple application examples were built and then tested by several blind users using a screen reader.

– *Method definition*. Lastly, the chosen criteria were classified in a systematic way, according to the usability definition, UI components and page elements involved.

The main navigational problems, which can potentially occur when using a screen reader or magnifier, include the following issues:

– *Information not self-explicative:* Reading through the screen reader or a magnifier, the user may lose the overall context of the current page and read only small portions of texts. For example, when skipping from link to link with the tab key, a blind user reads the link text on the Braille display, or hears it from the synthesizer, but does not know what is written before and after the link (e.g., ".pdf", "more details", etc.). A similar effect occurs when using a magnifier: at any given moment, only a small portion of the enlarged text can be visualized on the screen.

– *Information overloading*: The static portions of the page (links, frames with banners, etc.) may overload the reading through a screen reader, because the user has to read everything almost every time, thus slowing down the navigation. For instance, let us consider the case in which the user wishes to send an SMS message. After having filled in the form and clicked on the "send" button, the system provides a success or failure response. Often such result is visualized in the middle of the page, among other content. Therefore, it may take the user a long time to find it because the information that precedes the message has to be read, even if it is still the same as the previous page. Possible solutions to this issue include appropriate frame and link number usage, specific content marking, and a more organized hierarchical structure of pages.

– *Excessive sequencing in reading the information:* The command for navigating and reading can constrain the user to follow the page content sequentially. Thus, it is important to introduce mechanisms to ease the identification of specific parts in the page. An example of this is the result page generated by a search engine. Usually, in the top of such pages, there are several links, advertisements, the search fields and buttons, and so on, and then the search results begin. Furthermore, if the Web page contains more information blocks (e.g., paragraphs, short news, review lists, etc.), in order to read a specific block, the user has to read also the previous ones. An adequate partitioning or structuring of the content could allow users with visual impairments to find the desired information more quickly.

As result of this process, 18 criteria have been identified and grouped into three sub-sets. To codify each criterion we use the format i.j.l where: I denotes the criterion kind, that is 1 for

effectiveness, 2 for efficiency, or 3 for satisfaction; j is a progressive number to enumerate the criteria; l can be *a* (presentation) or *b* (dialogue) to indicate the interface aspect affected by the criterion.

3.2.1 Effectiveness criteria

We consider a criterion as belonging to the effectiveness sub-set if its application is important for reaching the user's goal. This means that if the criterion is not adopted, users may not be able to accomplish their tasks, because they will encounter difficulties in identifying important information.

Logical partition of interface elements (1.1.b). This criterion aims at grouping information, links, fields and so on in logical categories, in order to allow users who read the page content through special devices to localize the essential parts in the page. In fact, as Web pages usually contain a lot of data, often users are not able to find a specific part or piece of information. Moreover, some screen readers allow skipping from section to section. This can be obtained by using markers, frames or headings to group texts, links, forms, and so on, according to a logical division, for instance, by defining frames such as "index", "search" and "search results". If the Web site does not use frames, the same result can be obtained by marking each part with hidden labels having similar strings to those mentioned above.

In order to apply this criterion, developers can use various methods:

– Headings: applying appropriate tags <H1>...<H6>;

– Hidden markers: putting non-displaying descriptive labels before significant "blocks";

– Tables: assigning useful values to summary attributes;

– Frames: assigning proper strings to name and title attributes.

Proper link content (1.2.a). The link labels are important for special users who use screen readers and keyboard commands (e.g., the Tab key) and, for this reason, may not be able to see the overall context (i.e., text appearing before and after a link). For this reason, it is important that the links are clear and context independent, avoiding the use of general texts such as "more details", "download", ".pdf". We must warn the designer that such texts can lower the site's usability, because they are not unambiguous, or not informative enough. Therefore, designers can work on text, alternative text or TITLE attribute in order to improve the following aspects:

– text links: using more a meaningful description (i.e., text enclosed between <a> and), and/or the title attribute;

– graphical links: specifying a significant description for the alt attribute value, referring to the meaning of the link rather than describing the image in itself;

- graphical and text links: when both graphical and textual links to the same resource exist, both <img> and a text string should be enclosed between the <a> and tags, so that they are recognised as a single link by screen readers.

Messages and dynamic data management (1.3.b). A significant difficulty that users with vision impairments often encounter is represented by poorly designed system feedback. This may be confirmation or error messages about instructed operations or requested information, which are not presented in a manner that can be rapidly interpreted by screen readers, for example, messages in the middle of the page or amongst a lot of other information and links. In addition, these dynamic messages are often so "short" that they are not easy to focus on. This may actually lead to users wasting time or executing screen readers commands in order to reach these messages and, if they are successful, have the messages read. Developers have some possible ways to improve this situation:

- Focusing on new messages: including an anchor in correspondence to dynamic messages, and using a method to automatically position the browser on that anchor (e.g., by passing #anchor-name);
- Using a dialogue window: applying JavaScripts like "alert", "confirm" and "prompt";
- Opening a new window: using JavaScripts, such as 'window.open' with larger messages.

Loading suitable style sheets (1.4.a). Web browsers can load specific sheets for different items using a particular tag (@media types). This style sheet feature allows specifying how a document is to be presented on different media, such as screen, paper, speech synthesizer, Braille device, etc. This enables the definition of a style sheet for each requirement, thus improving the layout of Web pages. Therefore, it is useful to specify that a style sheet, or a section of it, applies to specific media types. In order to improve navigation of blind or visually-impaired people, voice synthesizer, display and printer Braille devices are taken into account by a specific section for each type of device.

Layout and terminological consistency (1.5.a). Layout and terminological consistency is a usability feature that allows users to better understand the context and the available functions. Button features have a very strong impact on reducing situations that may be unclear for users, especially when their overview of the content is restricted. It is important that all the pages of the whole Web site adopt the same labels for buttons performing the same tasks (e.g., OK/Yes, quit/exit, next/forward), and that all pages have the same layout (e.g., dimension, form and color). These two aspects have important consequences both for users of screen readers with speech synthesisers or Braille displays,

as well as for visually-impaired users, who rely mainly upon dimension/color references. Developers must carefully attend to some aspects:

– button labels: using the same labels (value attribute) for buttons having the same functions;

– button dimensions: applying the same dimensions (WIDTH and HEIGHT attributes) for all buttons;

– links to home pages: applying a standard terminology like "home page" or "home" (possibly including it in the title attribute).

3.2.2 Efficiency criteria

An efficiency criterion is a rule that allows users to find the desired information more quickly. We consider this rule less important than effectiveness, because if such criterion is not satisfied, users can still perform their task, although it may take more time.

Number of links and frames (2.1.b). It is important that a page does not contain too many links or frames, as this makes it difficult for the user to skim through them all. In fact, the number of frames and links in a Web page affects navigation through special devices. Pages should have neither too many nor too few links: lots of links take a long time for readers to get through, and too few links may imply too many levels in the Web pages' hierarchy structure. In defining the range of values, the possible types of Web site and page should be taken into account. For instance, for Web sites relating to online books or guides, few frames are advisable (e.g., the navigation bar, index and content), while for Web sites providing several services, more frames can be useful to distribute the various information and services. A similar rationale can be applied to the number of links: it is advisable to keep the number of links down in pages containing mostly reading material (paragraphs, chapters or papers), whereas in pages with indexes, search results or lists of services, more links may be necessary. Concerning Web sites structured in frames, in judging the number of links to use, we suggest considering each frame as a single page. In a page having both an index and a content frame, the index can have numerous links, whereas their number in the content page should be limited.

– Number of frames: considering suggested ranges according to type of Web site;

– Number of links: considering advisable number of links on the basis of Web page categories.

Proper name for frames, tables and images (2.2.b). It is important to insure that all frames, tables and images have names and descriptions which are appropriate and meaningful, e.g., frames with names such as "top frame", "mid frame or "left frame" are not very helpful to the user. On the other hand, names such as "index", "search" and "content" can make it easier for users to reach

their goals, because it gives them the option to skip some frames and cut down the amount of information to read. Similar considerations apply to text summaries of tables and to alternative description for images and graphics. Descriptions such as "this is a layout table" or "logo.jpg (100 kb)" are of no use to users, who must listen to extra text without the benefit of receiving significant information. To summarize, the relevant suggestions are:

– Frames: using meaningful values for the name and title attributes;

– Frames: careful consideration of the <title> tag of pages loaded into frames;

– Tables: providing a meaningful description through the summary attribute;

– Image description: using the alt and longdesc attributes to indicate the type of picture in question (e.g., logo, decorative, link use, and so on).

Location of the navigation bar (2.3.a). The so-called navigation links (i.e., the links appearing on each page and enabling users to reach the main parts of the site) represent a source of delay and inefficiency for screen reader users. Since such links appear on each page (and often even twice), the users who are forced to read the contents in an almost sequential way (by means of a speech synthesizer or a Braille display) are always compelled to skim them before they can identify the contents of the current page. Therefore, highlighting the navigation bar at the top and/or the bottom of the page, if any, can be useful to make it more understandable to users who are unable to see its visual features (e.g., the horizontal or vertical position, the color or font types, etc.) and can increase navigation efficiency for these users. Other features can be used in order to localize the navigation bar through a screen reader. Therefore, we need to select criteria that makes navigation easier for people using a speech synthesizer (e.g., by graphic and/or text references or frames), and, on the other hand, for visually-impaired people (e.g., by different colours and dimensions). The suggestions are as follows:

– Hidden markers: putting specific hidden text in order to mark the start of the navigation bar;

– Frames: defining a specific frame dedicated to navigation links, such as "navigation" or "navigation bar";

– I-frame: defining a frame reserved for navigation bar, also when frames are not used for all Web sites;

– Menus and submenus: using list tags (e.g., <ul>) in order to obtain a hierarchical structure easily identifiable with the screen reader; applying font and color features in order to visually differentiate multilevel menu elements;

– Popup menu: using appearing or disappearing blocks by means of <div> tags appropriately organized in sequential order, and by setting the visibility properties.

Importance levels of elements (2.4.b). In order to facilitate navigation, especially when using the keyboard, it is possible to assign different importance values to interaction elements such as links, buttons, and fields. This way, when users move with the "Tab" key (element by element), they visit at first the most important elements, and later the less important elements, regardless of their location on the page.

One possible application of this criterion is to assign higher importance values to the elements of the current page that are new with respect to the previous page in order to make their access easier.

Another example of this application is when filling in a form, it would be useful to reach, through the Tab key, first the obligatory fields and then the optional ones. Links belonging to the navigation bar could have a lower level, whereas those of the current page could have a higher level. Therefore, developers have to decide how to assign the various levels. Our suggestion is to use tabbing order position: using the TABINDEX attribute to assign different importance levels to interaction elements (the navigation goes from the lowest tabindex value to the highest one).

Assignment of shortcuts (2.5.b). It is advisable to assign hot keys to the most important buttons, links and fields, so that the user is able to reach them quickly through a simple key combination. This feature may be useful especially when users visit the Web site frequently, and learn the key combinations by heart. In addition, a specific (possibly hidden) link to a page of combination list should be added to the navigation bar. Note that we classify this aspect among the "efficiency" criteria because it enables reaching an object more quickly, but we could also include it in the "satisfaction" group, mainly for those users accustomed to using a lot of key combinations. The suggestions are:

– Main links, buttons and fields: applying the accesskey attribute with mnemonic shortcuts (usually the first letters);

– New interaction elements: using same accesskey value (e.g., the letter "n") to skim through new elements.

Proper form layout (2.6.a). In forms with several groups of data, we must properly lay out group titles and fields to achieve a major clearness. In fact, the way the elements are formatted can cause confusion with the screen reader. For example, in some cases the voice synthesizer or Braille display could read before the "checkbox", "combobox" or "field" item, and after its value, or vice versa. Often blank characters can cause difficulties in reading if not appropriately used. Thus, a correct application of layout elements (e.g., simply by using the return tag in the proper place) is recommended.

- Button labels: giving a text to the value attribute;
- Groups of control elements: appropriately combining the <fieldset>, <legend> and <br> tags;
- Onchange event: the use of onchange event can create difficulties for users, especially when navigating through the keyboard; other elements (e.g., the onclick event) can be used instead to obtain a similar effect;
- Matching labels and input elements: using the <label for> tag in order to associate correctly each field label with its input element (this is especially useful when layout tables are used).

Specific sections (2.7.b). In sites with frequent information updates and/or new resources to download, we can help the user to find more rapidly the new elements by providing a specific section listing the new elements by date, sparing the user the trouble of going all over the site. Furthermore, a specific page listing all short keys associated to the most important links of the Web site should be considered.

- A "Last update" section: creating a specific Web page containing the list of recent updates, in inverse chronological order (so that users read first the more recent updates); for each update a link to the real updated Web page should be provided. Then, a specific link to the updates Web page would be added to the navigation bar or to the Home page;
- A list of assigned keys: defining a specific page containing shortcut list used in the whole Web site, and adding to the navigation bar a (possibly hidden) link to that page.

Indexing of contents (2.8.b). In pages containing information of different kinds (paragraphs, news, etc), we can help the user to find information more efficiently by indexing and pointing out different blocks, avoiding the need of page skimming. First of all, it is useful to logically organize the information (by using headings, markers or frames) and then apply a kind of index (i.e., local links) pointing to every "section". The suggestions are:

- Highlighting each block: to this end it is possible to insert a link that is not displayed but can be detected and interpreted by the screen reader. The title block can become a sham link: this can be obtained through the insertion of an anchor at the beginning of the text following the title, and then the creation of a link with the title text, pointing to that anchor (so, moving through with the Tab key, the screen reader can detect all titles);
- Index at the top of page: it can be done by putting an anchor in correspondence of each block, and then adding a list of links at the beginning of the page (possibly enclosing links in a specific i-frame);
- Index frame: if the Web site uses frames, a specific frame containing the link to every section can be inserted.

Navigation links (2.9.b). In order to reach more easily some location in the page (or in the site) we can insert local navigation links referring to bookmarks in the scope of the page (e.g., "skip to content", "go to top", "go to navigation bar", etc.). Links of this kind are useful for navigation through screen readers. The suggestions are:

– Skip to content: including an anchor in correspondence of the new page content (i.e., after the navigation bar or advertisement data) and adding a local link to that bookmark at the top of the page;

– Go to the top of the page: inserting a link that allows going back to the beginning of the page, the bottom of the page, or after each section (if any);

– Go to specific section: relating to some special sections (e.g., the navigation bar, the search field, the index, and so on), defining specific links at the bottom or the top of the page.

Identifying the main page content (2.10.b). One of the most important consequences of blind users' not being able to view a page overview, is that they are unable to identify new content on a page. In fact, Web pages often have diverse information, objects, elements and so on, which are static, and only a part changes when a new page is loaded. The modified part can be called, for example, "new page content". Blind users may not be able to identify the "new page content", because the screen reader deals with the page sequentially, and hence too much data (sometimes poorly organized) may create confusion and render the content unclear. For this reason, we suggest some possible strategies to use in order to address such issues.

– Positioning at page loading: the focus is automatically moved to the new content when the page is loaded;

– Using heading levels: the first line of new page content (usually its title) should be the first text to which a heading level <hi> tag has been applied;

– Using frames: when the Web site has a frame structure, a specific frame named 'content' should be used.

3.2.3 Satisfaction criteria

Satisfaction criteria help to produce a Web site that is more pleasant and easier to navigate. Often, the application of satisfaction criteria provides useful information without the need of any special command. For example, applying a short sound to the page loading means providing an additional piece of information that allows users to know, exactly and in real time, when the page is completely loaded. When no sound is applied, users can still accomplish their task, but doing that will probably require more effort.

Addition of short sounds (3.1.b). Associating a short sound to different elements and different kinds of a multimedia environment can make the user more "satisfied"—e.g., providing each page with a short sound indicating when the loading of the page is completed eliminates the need for the user to repeatedly check the status bar. Associating different sounds to different links makes it easier to identify the link type during skimming.

Colour of text and background (3.2.a). This aspect can make navigation easier for visually-impaired people who, with a particular type of contrast, may feel less tired by navigation. It is therefore advised to avoid color combinations giving a poor contrast. Furthermore, changing colors in correspondence to some events, or particular areas, can be a way to get attention. The suggestions are:

– By moving the mouse: using CSS specifications (or JavaScripts) to change color when the mouse pointer is moved onto elements (links or buttons);

– Getting focus: using CSS specifications (or JavaScripts) to change color when elements are in the focus;

– Specific areas: applying different background and text colors for particular areas (e.g., navigation bar or index frames, search or result area, and so on).

Magnifying at passing by mouse (3.3.a). The use of this feature can help people with a good visual residue to better focus on the pointed object. The idea is to enlarge particular elements such as images, navigation links and buttons, not all text. We suggest magnifying the elements only when the mouse pointer goes over them, because this feature can be useful for users who have a visual residue and therefore can use the mouse.

– Images: using JavaScripts to associate an "enlarge" function when the picture is pointed (onMouseOver) and a "reduce" function when the pointer goes out (onMouseOut);

– Links and buttons: using CSS instructions (or JavaScripts) to magnify the elements when the mouse is moved onto them.

3.3 The impact of the criteria on the Web pages code

Different criteria have different impact on the Web pages code. Some of them require changing specific elements such as the links or the frames. Others have an impact on the pages structure, and others have an impact on the entire site. Table 1 summarises the impact of each criteria.

4 Some examples

In this section we show some examples of usability issues in accessible Web sites detected when users interact through the screen reader Jaws. The examples are taken from three Web sites, evaluated by us through the set of proposed criteria. First, we considered the Web site of the Florence City Council (http://www.comune.firenze.it), which provides information and services to citizens, paying particular attention to the social security department (i.e., "The services" section). Then, we tested a Web site on public access (http://www.pubbliaccesso.it), which aims at providing both developers and users with information regarding accessibility topics. Lastly, the third Web site we analyzed is about the Ministry of Environment (http://www.minambiente.it/). More precisely, in this case, an accessible prototype has been developed by another group for supporting the search of information concerning public and private institution. After that portion has been made accessible according to the WAI W3c guidelines, we applied our criteria.

These studies have been done to assess how developers deal with usability when developing accessible Web sites, and how the criteria adopted are suitable to be applied to existing Web sites. For each example, we show how the page is visualized on the screen, how that page is read by the screen reader before and after our suggested changes, and the code fragments involved. In the table showing how the Web page is interpreted by the screen reader, the bold text with [1] indicates changes, while the italic text with [2] refers to the parts that are read by the synthesizer, but not visualized in the Web page.

4.1 Proper text: frames, links and tables

An important issue of usability for users who read the Web pages through a screen reader is the text associated to links, the name attribute of frames, the summary attribute of tables, and so on. The reason is that, while exploring a page with a synthesizer or Braille display, the user may not have a general view of the content. Therefore, if the name of frames or the text of links are significant, users can better orient themselves.

4.1.1 Names of frames

Unfortunately, often the names of frames are not much meaningful, such as sx, dx, central, main, etc. This is particularly important when users read with a screen reader. For better usability, the name of a frame is important for an easier understanding of both the page structure and the frame

content. Usually, this is not important, because the names of frames are not shown in the page visualized in the Web browser and therefore developers do not pay attention to them.

Let us consider the Web page in Fig. 1, where services offered by the City Council of Florence are listed. The structure of that page is composed of several frames which are not entirely visible.
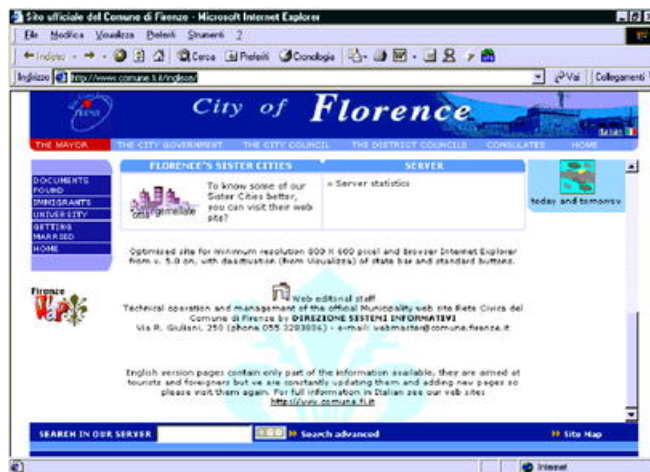

Fig. 1. Services offered by the City Council of Florence

In that page information and links to skip to other sections are logically grouped in more nested frames. The picture shows how the page content is rendered on the screen. A screen reader reads that content in different ways. In order to try to understand how the synthesizer or the Braille display considers the page, we provide a fragment of the Web page text read by the synthesizer (see Fig. 2). As we can observe, the screen reader identifies clearly the beginning and the end of every frame; thus, the use of appropriate names is important.

Fig. 2. A fragment of the Web page text read by the synthesiser

As we can see in the figure, the screen reader distinguishes the frames of the page and also their beginning and ending. Therefore, it is useful to have significant names. Note that the "main" frame is not very important, because it is actually used to contain the others.

A similar situation occurs also in the "ricerca enti" (organisations search) frame of the Ministry of Environment Web site. In that section there are five frames that originally had not very appropriate names (see Fig. 3).
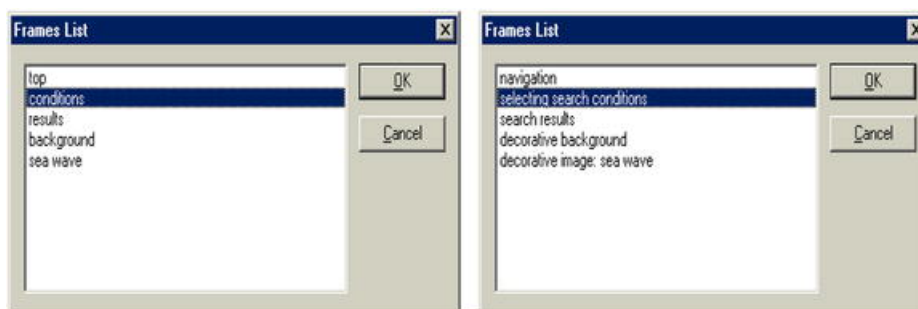
Fig. 3. Inappropriate frame names

The user can read the content of the page in a sequential way, or skip directly to a certain frame. In fact, some screen readers (like Jaws) have special commands to list the available frames, and to activate one of them. The user can choose one frame to access it. For instance, selecting "search results", users access the corresponding frame so that they are able to find the search results more quickly.

In order to show the code related to the frame names, we consider a fragment of the page of the Florence site (Fig. 4). In particular, we consider the frame tags: first of all, the presence of both name and title attributes, then their content.

```
<HTML>…<frameset cols="*,750,*">
<frame name="main"¹ src="main3ing1.htm" title="main"¹
frameborder="0"> <noframes>…</noframes> </frameset> </HTML>
<HTML>…<frameset rows="84,*">
<frame name="navigation bar" title="navigation bar"¹ frameborder="0"
scrolling="NO" noresize src="top5ing.htm"> <frameset rows="*,29"
> <frame name="content" title="content"¹ src="home3ing.htm"
scrolling="yes" frameborder="0">
    <frame name="search bar" title="search bar"¹ scrolling="NO" noresize
src="bottom2ing.htm" frameborder="0"><noframes>…</noframes>
</frameset> </frameset> </HTML>
```

Fig. 4. A fragment of the Florence Web site

Furthermore, it is important that the strings associated to name and title attributes are the same. In fact, if the values are different, the screen reader could behave in different ways, depending on the type of interaction technique used (i.e., it reads the name if the user interacts through the Ctrl-Tab browser command, and the title if the user uses the screen reader commands).

In order to evaluate the frame name, we could use a dictionary in which the evaluator can store unsuitable terms, such as sx, dx, central, main, etc. Using an external dictionary we can have two advantages: firstly, the evaluator can add new terms customizing the evaluation, and secondly, there may exist different dictionaries for different languages.

4.1.2 Link content

A similar problem occurs with links like "click here", ".pdf", or "go to paragraph", which are not very useful. Often the link text refers to a specific context, so that if we consider links separately, we might not be able to understand the related content.

A user who cannot see the screen often uses the Tab key to search in the page the link wanted without reading the whole content. Another way to select a specific link is to use a particular command of the screen reader to open the link list. In both cases, the user reads only the text of the links, so a meaningful text is important.

Figure 5 shows an instance of this issue. In the pictures there are the lists of links associated to each topic belonging to an online guide organized in more paragraphs. There is one graphical link before every paragraph item. Since the links are images, the use of alt is necessary. In the original version all links have the same text: "go to paragraph". In the fixed version, the alt attributes contain also the name of paragraph (e.g., "go to paragraph presentazione").
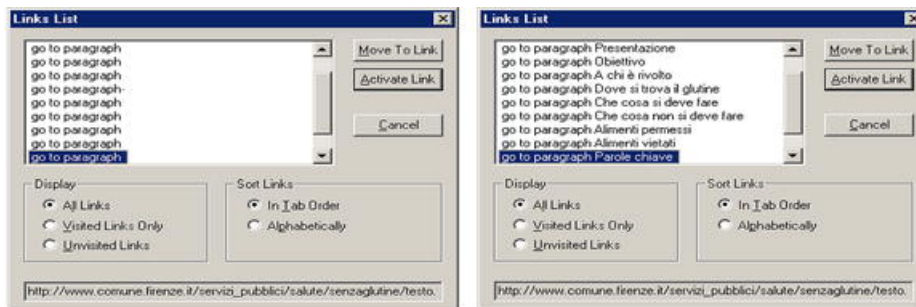
Fig. 5. Lists of links associated to each topic belonging to an online guide organised in more paragraphs

In order to improve the link usability, we can suggest some solutions. Firstly, for graphical links, we have to apply the alt attribute with a meaningful description which has to refer to the meaning of the link rather than describing the image in itself. Secondly, for textual links we can change the entire text, or we can use alt or title attributes. This second alternative may be used if developers do not want to modify the writing visualized on the screen. In fact, the text associated to alt or title is read by the screen reader, and it is visualized in the status bar when the mouse is passed over links.

Reading the Web page by a screen reader, the original links are like "*link*[2] go to paragraph[1]", while those modified are like "*link*[2] go to paragraph presentazione[1]". (Fig. 6) it can be easily noticed that in the original page there are too many links with a similar text, such as "go to paragraph". In the presence of many links of this kind, if users skip from link to link using the Tab key or a special command of the screen reader that gives a link list, they read similar texts without knowing the context to which they refer. Therefore, in order to know which is the appropriate link, a user has to explore the page reading almost line by line.

```
… <A target=content
href="http://www.comune.firenze.it/…/testo.htm#presentazione">
<IMG height=19 alt="go to paragraph Presentazione"¹
src="sommario_file/spunta2.gif" width=20 border=0></A>
Presentazione <BR> <A target=content
href="http://www.comune.firenze.it/…/testo.htm#obiettivo">
<IMG height=19 alt="go to paragraph Obiettivo"¹
src="sommario_file/spunta2.gif" width=20 border=0></A>
Obiettivo <br>
...
```

Fig. 6. Links read with a screen reader

In order to solve this problem, we should modify the text of the links by adding the name of the chapter to which the link points (e.g., "presentazione", "obiettivi", etc.) to "go to paragraph". This can be obtained by changing the link text, or using the alt and title attributes. This second possibility, among other things, allows two renderings: a visual and one for the screen reader. In our example, the links are images, therefore we have to modify the alt and title attribute text: on the screen the links are still images, while the screen reader reads the text "go to paragraph presentazione", etc.. In addition, for each link, the evaluation has to check whether the text is non-recommended or not. Similarly to frames, we assume that external dictionaries in which non-recommended terms are listed exist. The evaluation criterion extracts text of links—including ALT and TITLE value—and checks that they do not belong to that terms.

4.1.3 Tables: proper summaries

Using tables to format the content in the pages is usually not recommended. However, developers often prefer to use this technique because it allows obtaining more of a precise content layout. Therefore, in this case our criteria suggests defining useful strings for the summary attribute. Let's consider now the example depicted in Fig. 7 in order to give an example about this aspect. That page, structured by using frames, has a specific frame named "Selection search conditions" where more tables are used in order to format the search criteria.
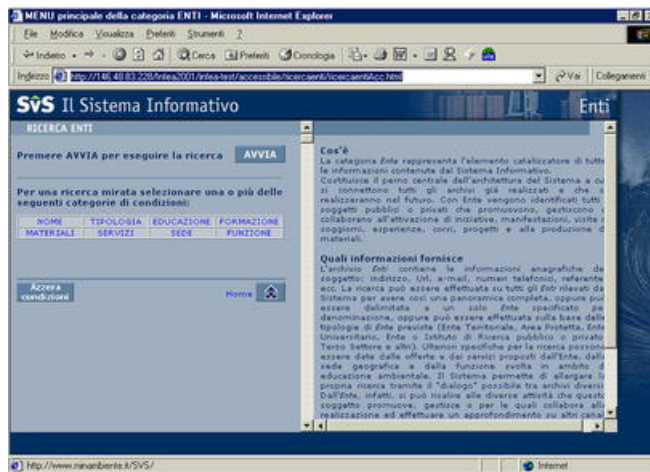
Fig. 7. Search criteria with frames

In particular, in addition to the "search parameters" table (see Fig. 7) where there are local links to move the focus to a specific criterion, a layout table for each search criterion is used. Actually, several nested tables have been used to visualize checkboxes, combo boxes, radio buttons and text fields that are shown when the associated link is selected. Because too many nested tables have been used, it was not easy to appropriately modify the corresponding code. Figure 8 shows how the screen reader interprets that content, before and after our suggestions. As before, the italic text with 1 indicates changes, while the italic text with 2 refers to the parts that are read by the synthesizer, but not visualized in the Web page. (Fig. 9 shows the selection conditions via the screen reader).

Fig. 8. How the screen reader interprets nested content

Fig. 9. Selection conditions via the screen reader

Practically by simply pressing the letter "t", the screen reader skips table by table, reading the type of selection category each time (e.g., by name, by typology, by education, and so on). In order to obtain such an effect, the page code should contain a <table> tag with an appropriate summary attribute for every category of selection.

4.2 The navigation bar and menus

Pointing out the navigation bar, if there is one, at the top and/or the bottom of the page, can be useful for users who are not able to see its visual features (e.g., the horizontal or vertical position, the color or font types, etc.). Other features can be used in order to localize the navigation bar by a screen reader. We could mark the beginning of the static links, insert a particular link to skip the bar, and so on. Specific labels and strings can be used in order to identify the navigation bar. In the next two paragraphs, two examples of this issue are described.

4.2.1 The navigation bar

Let's consider the page in Fig. 10 where a specific section of the PubbliAccesso Web site is visualized. Two navigation bars are used: one at the beginning, and one at the end of the page. In addition, a hierarchical structure menu is available on the left (see the next paragraph).

Fig. 10. A specific section of the PubbliAccesso Web site

To format the navigation bars, layout tables have been used. In this case we suggest the application of an appropriate summary attribute to those tables, such as "Navigation bar" instead of the misleading "Right part of the head". This way, users are able to immediately understand the goal of the table. However, we suggest inserting a link like "skip to content" in order to avoid reading the navigation bar content (which is always the same) and to go directly to the new content. In the next two figures (Fig. 11 and Fig. 12), we can see how users read the page content (i.e., how they can identify the navigation bar) and then which is the code involved. (Fig. 13 shows the Webmaster public access page).

Table with 2 columns and 5 rows[2]
Summary[2]: **Right part of the head**[1]

Link[2] `Utenti`
Link[2] `WebMaster`

Link[2] `Cerca |`
Link[2] `Personalizzazione |`
Link[2] `Mappa`
table end[1]
`… Page content …`
Link[2] **Skip to content**[1]
Table with 2 columns and 5 rows[2]
Summary[2]: **Navigation bar**[1]

Link[2] `Utenti`
Link[2] `WebMaster`

Link[2] `Cerca |`
Link[2] `Personalizzazione |`
Link[2] `Mappa`
table end[1]

Link[2] **Skip to content**[1]

Graphic[2] **Navigation bar:**[1]
Link[2] `Utenti`
Link[2] `WebMaster`

Link[2] `Cerca|`
Link[2] `Personalizzazione|`
Link[2] `Mappa`

Graphic[2] **Page content**[1]

Fig. 11. How readers can identify the navigation bar

```
<TABLE style="WIDTH: 100%" cellSpacing=0 cellPadding=0
border=0 summary="Right part of head"[1]>
<TBODY>
...
<TD class="" style="BACKGROUND-COLOR: #297618">
<A class=menu_bar title="Go to user section"
href="http://www.pubbliaccesso.it/sector.asp?idsezione=119">Ut
enti</A>
</TD>
<TD class="" style="BACKGROUND-COLOR: #297618">
<A                      class=menu_bar title="Go to
Webmaster section"
href="http://www.pubbliaccesso.it/sector.asp?idsezione=120">We
bMaster</A>
...
</table>
```

```
<a href="#content" title="skip to content"> Skip to content </a>[1]
<TABLE style="WIDTH: 100%" cellSpacing=0 cellPadding=0
border=0 summary="Navigation bar"[1]>
<TBODY>
...
<TD class="" style="BACKGROUND-COLOR: #297618">
<A class=menu_bar title="Go to user section"
href="http://www.pubbliaccesso.it/sector.asp?idsezione=119">Ut
enti</A>
</TD>
<TD class="" style="BACKGROUND-COLOR: #297618">
<A                      class=menu_bar title="Go to
Webmaster section"
href="http://www.pubbliaccesso.it/sector.asp?idsezione=120">We
bMaster</A>
...
</table>
...
<a name="content"> </a>[1]
```

```
<a href="#content" title="skip to content"> Skip to content </a>[1]
<img src="" alt="navigation bar:" width=1 height=1>[1]
...
<A class=menu_bar title="Go to user section"
href="http://www.pubbliaccesso.it/sector.asp?idsezione=119">Ut
enti</A>
<A                      class=menu_bar title="Go to
Webmaster section"
href="http://www.pubbliaccesso.it/sector.asp?idsezione=120">We
bMaster</A>
...
<img src="" alt="Page content" width=1 height=1>[1]
<a name="content"> </a>[1]
...
```

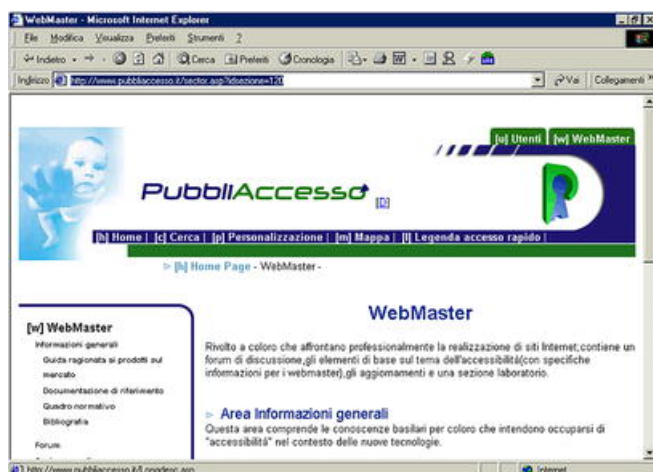**Fig. 12.** Which code is involved in how readers read page content

Fig. 13. Webmaster public access page

So, we have provided two ways to identify a navigation bar: the first one uses a table with an associated specific summary attribute (i.e., a navigation bar); the second one applies hidden labels to mark the beginning of the navigation bar and of the page content. Both ways provide a specific link to skip to content that, if chosen, frees the user from having to read the navigation bar links.

4.2.2 Menus and submenus

When in a Web page there are several links organized into various levels, lists can be used. When we build a menu composed of sub-items, it is important to distinguish which are main items and which sub-items. An unordered list (i.e., <UL>) can solve this issue.

This paragraph provides an example of the above. In the PubbliAccesso Web site, a multilevel submenus is positioned to the left side of the page.

In order to point out the hierarchical structure of that multilevel menu, two different methods are applied: with CSSs, different font sizes and indenting allows visually impaired users to understand the hierarchical structure; by using <ul> tags, blind users are able to get information on multilevel menu (i.e., items and sub-items). The fragment of code is shown in Fig. 14. The part enclosed between <style>and </style>is referred to visual features, whereas that between <ul> is useful for screen reader interpretation.

```
<STYLE type="text/css">
...
A.menu { FONT-SIZE: 112% }
A.sub_menu { FONT-SIZE: 78% }
A.menu { FONT-WEIGHT: bold; COLOR: #000000;
         TEXT-DECORATION: none }
A.sub_menu { FONT-WEIGHT: bold; COLOR: #000000;
             TEXT-DECORATION: none }
A.menu:hover { TEXT-DECORATION: underline }
A.sub_menu:hover { TEXT-DECORATION: underline }
UL.menu { MARGIN-BOTTOM: 10px; MARGIN-LEFT: 10px;
          LINE-HEIGHT: 140%; LIST-STYLE-TYPE: none }
...
</style>
<body>
...
<UL class=menu>¹
<LI><STRONG><A class=menu title=""
href="http://www.pubbliaccesso.it/…">Users</A></STRONG>
<UL class=menu>¹
             <LI><A class=sub_menu title="Vai all'area Forum-
Utenti"
href="http://www.pubbliaccesso.it/forum_utenti/…">Forum</A>
             <LI><A class=sub_menu
             title="Vai all'area Elementi di base-Utenti"
ref="http://www.pubbliaccesso.it/sector.asp…">Elementi di
base</A>
...
</ul>¹
<LI><STRONG><A class=menu title=""
href="http://www.pubbliaccesso.it/sector.asp…">WebMaster</A></ST
RONG>
...
</ul>¹
</body>
```

Fig. 14. The fragment of code showing hierarchical menu structure

As we can observe in the next figure, the voice synthesizer (or Braille display) reads the hierarchical menu that is defined by "List of 2 items", "List of 5 items nesting level 1", where "nesting" indicates the submenu. We marked the bolded portions by *1,2* because those parts are both read by the screen reader (and non-visible on the screen) and involved by criteria modifications (i.e., the result of <ul> tags). (Fig. 15 shows nesting level items).

List of 2 items[1,2]
- *Link*[2] Utenti

List of 5 items nesting level 1[1,2]
- *Link*[2] Forum
- *Link*[2] Elementi di base
- ...

List end nesting level 1[1,2]
- *Link*[2] WebMaster

List of 4 items nesting level 1[1,2]
- *Link*[2] Informazioni generali
- *Link*[2] Forum
- ...

List end nesting level 1[1,2]
List end[1,2]

Fig. 15. Nesting level items

4.3 Identifying information and elements

Structuring content and elements, so that users can be able to find more quickly the information wanted, is an important goal of our criteria. Examples of solutions of this issue are: logical organization of the information and association of shortcuts or indexing levels to elements.

4.3.1 Using headings

A useful method by which information can be logically organized is to use appropriate heading elements. Earlier we discussed using appropriate layout tables by associating them to meaningful summary attributes. The same result can be obtained by using heading tags. In fact, several screen readers have special commands to list the headings in the page, so that users are able to understand, with a general overview, which is the content of the page, and can reach some information more quickly. Fig. 16 depicts heading use applied to the Ministry of Environment Web site: a single heading has been used for each selection category, as when using table summary attributes. We evaluated that also in the PubbliAccesso Web site headings were applied correctly (i.e., to logically group information).
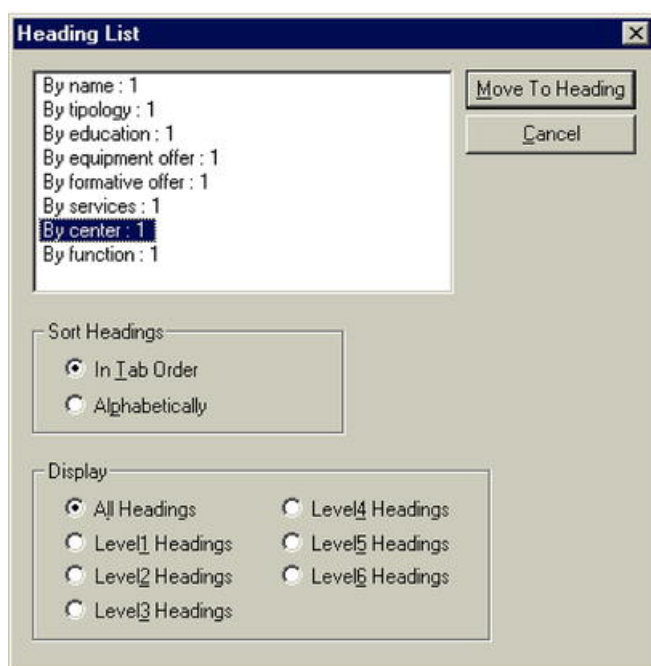
Fig. 16. Heading use applied to the Ministry of Environment Web site

In Fig. 17, the HTML code which creates the layout tables of the abovementioned example is shown. Practically, <h1> tags have been used with image tags: the screen reader produces heading list getting values in alt attributes. By associating these tags with image tags, there is no effect on the screen: they are interpreted only by the screen reader.

```
<table width="330" border="0" cellspacing="0"
cellpadding="0"><!--INIZIO TABELLA FILTRI NOME-->
<tr> <td colspan="2" valign="middle" height="55">
<h1> <img src="../Denominazione.gif" height="16" width="330" alt="By name">
</h1>
</td> </tr>

...
<table width="330" border="0" cellspacing="0"
cellpadding="0"><!--INIZIO TABELLA FILTRI TIPOLOGIA-->
<tr> <td colspan="2" valign="middle" height="55">
<h1><img src="../Tipologia.gif" height="16" width="330" alt="By
typology"></h1>
</td> </tr>

...
<table width="330" border="0" cellspacing="0"
cellpadding="0"><!--INIZIO TABELLA TITOLO OFFERTA_EDUCATIVA--
>
<tr> <td valign="bottom" height="55">
<h1><img src="../OffEdAcc.gif" height="16" width="330" alt="By
education"></h1>
</td> </tr>
...
```

Fig. 17. The HTML code which creates the layout tables

4.3.2 Shortcuts and indexing levels

Shortcuts and indexing levels associated to interaction elements are two useful tools for users

navigating mainly through the keyboard. In fact, these features allow users to find and reach more

quickly the wanted elements. We suggest applying shortcuts only to main links (e.g., those of the

navigation bar) and not to all the links; otherwise, users cannot remember them, and shortcuts are

not useful anymore. In addition, it would be better if the assigned letters were the first of the words

they are referring to. The PubbliAccesso Web site has assigned shortcuts to the navigation bar

links shown in Fig. 10, picking the first letters of the link texts. To assign a shortcut to a link, the

accesskey attribute can be used as reported in Fig. 18.

```
<TD class="" style="BACKGROUND-COLOR:#297618"><A
class=menu_bar title="Vai al settore Utenti" accessKey="u"
href="http://www.pubbliaccesso.it/..."
tabindex="10">[<B>u</B>] Utenti</A></TD>
...
<TD class="" style="BACKGROUND-COLOR: #297618"><A
class=menu_bar title="Vai al settore WebMaster" accessKey="w"
href="http://www.pubbliaccesso.it/…" tabindex="10"> [<B>w</B>]
WebMaster</A> </TD>
...
<TD style="COLOR: #ffffff; BACKGROUND-COLOR: #211771"
colSpan=1> <A class=menu_bar title="Torna all'Home Page"
accessKey="h" href="http://www.pubbliaccesso.it/…"
tabindex="10"> [<B>h</B>] Home</A> |  
<A class=menu_bar title="Vai alla pagina di ricerca"
accessKey="c" href="http://www.pubbliaccesso.it/cerca.asp"
tabindex="10"> [<B>c</B>] Cerca </A> |  
<A class=menu_bar title="Vai alla personalizzazione"
accessKey="p" href="http://www.pubbliaccesso.it/…"
tabindex="10"> [<B>p</B>]
...
<DIV class=sezione>
<A class=sezione href="http://www.pubbliaccesso.it/..."
tabindex="5"> Area Informazioni generali </A> </DIV>Questa
area comprende le conoscenze basilari...<br>
<DIV class=sezione>
<A class=sezione href="http://www.pubbliaccesso.it/..."
tabindex="5"> Area Forum </A> </DIV>L'area contiene... <br>
```

Fig. 18. Accesskey attributes

In order to inform users of the hot keys available during navigation, we advise providing a specific section entitled "list of available shortcuts", and adding a link to the navigation bar. We do not suggest indicating the hot key with the element name (e.g., [s] Search), because screen readers are able to recognize the accesskey value. Consequently, in this case, users would read the associated letter twice, which would be a cause of confusion. Figure 19 depicts this situation: every link has the indication of both "[letter]" (at the beginning) and "alt+letter" (at the end).

Fig. 19. Navigation bar links

Another tool that allows moving more quickly on the interaction elements is the indexing level use. We suggest assigning higher values to navigation bar links, and lower values to other links in the current Web page. In this way, users (through the "Tab key") visit the new elements first, then those of the navigation bar. Specifically, in our case, users first move to "Link Area Informazioni generali", then to "Link Area Forum", and so on (see Fig. 19). Subsequently, through the "Tab key", the focus is placed on the first link with the next lowest value, that is, "Link [u] Utenti", "Link [w] WebMaster alt+w", and then on all the others. The code to implement different indexing levels is shown in Fig. 18. Concluding, shortcuts allow users to immediately reach one element, whereas the indexing level enforces a certain "order" in the interaction element navigation.

5 Conclusions and future work

In recent years, there has been an increasing interest in accessibility and usability issues, because it is more and more important that the information be easy reachable by all. In this paper we have considered both concepts, and in particular we have considered usability for accessible Web sites. Therefore, we propose criteria in order to improve user navigation and remove potential usability problems when reading the Web pages using special devices. We have identified eighteen usability

criteria for accessible Web sites, which we partitioned in three sub-sets according to usability aspects: effectiveness, efficiency and satisfaction.

The proposed criteria can be used by developers, on the one hand, in order to ensure usability for disabled users; on the other hand, to evaluate the usability of accessible Web sites. Some examples that show how criteria can be applied and how they can be used to improve a site have been reported. Since evaluating and repairing Web sites by using the criteria requires several efforts and a lot of time, a tool supporting this activity can be a valid help for evaluators and developers. Thus, we have started the design and implementation of a tool supporting the criteria discussed in the paper.

References

1. Ivory MY, Hearst MA (2001) The state of the art in automating usability evaluation of user interfaces. ACM Comp Surv 33(4):470–516

2. Rowan M, Gregor P, Sloan D, Booth P (2000) Evaluating Web resources for disability access. In: Proceedings of the Fourth Annual ACM Conference on Assistive Technologies (ASSETS00), Arlington, VA, 13–15 November 2000, pp 80–84

3. CAST: Center for Applied Special Technology (2004) Bobby accessibility evaluation tool for Web sites. http://www.cast.org/bobby. Cited 2004

4. UsableNet (2004) LIFT, an accessibility evaluation tool. http://www.usablenet.com/lift. Cited 2004

5. Nist Web Metrics (2004) http://zing.ncsl.nist.gov/WebTools/tech.HTML. Cited 2004

6. Stephanidis C, Akoumianakis D, Sfyrakis M, Paramythis A (1998) Universal accessibility in HCI: process-oriented design guidelines and tool requirements. In: Stephanidis C, Waern A (eds) Proceedings of the 4th ERCIM Workshop on User Interfaces for All, Stockholm, Sweden, 19–21 October 1998

7. Grammenos D, Akoumianakis D, Stephanidis C (2000) Integrated support for working with guidelines: the Sherlock guideline management aystem. Int J Interact Comp 12(3):281–311

8. WAI Accessibility Guidelines (1999) Web Accessibility Initiative, World Wide Web Consortium, 1999. http://www.w3.org/wai. Cited 1999

9. Leporini B, Paternò F (2002) Criteria for usability of accessible Web sites. In: Proceedings of the 7th ERCIM Workshop. User Interfaces for All, Paris (Chantilly), France, 23–25 October 2002