

EFFECTIVENESS, PRODUCTIVITY AND SATISFACTION OF PERSONS WITH SIGHT AND MOTOR DISABILITIES WHEN USING DYNAMIC TEXT-ONLY PAGES

GIORGIO BRAJNIK^a and DANIELA CANCELILA

*Dip. di Matematica e Informatica
Università di Udine — Udine, Italy
giorgio@dimi.uniud.it
cancelila@dimi.uniud.it*

Received July 25, 2005
Revised December 29, 2005

Text transcoders are web-server systems that produce, on the fly, a text-only version of a web page requested by a user of a browser. Although the potential benefits of text transcoders are multifaceted and discussions on appropriateness of text transcoders to produce accessible versions of web sites are still ongoing, at the moment the impact of transcoded pages on disabled web users has not yet been scientifically studied.

This paper describes an experiment aimed at evaluating usability of web pages processed by a text transcoder and used by 29 disabled persons. Results based on subjective and objective data show how usability changes, and which results can be generalized to a broader population.

Keywords: usability, accessibility, text transcoder, text-only page

1 Introduction

Transcoders are proxy-like systems that respond to requests sent by the user's browsers; they relay these requests to an ordinary web server, collect the requested pages, transform and finally return transformed pages to the browser. In particular, they strip images, multimedia objects, JavaScript code from the page, and change its layout.

Text transcoders have been promoted as tools that can automatically yield text-only versions of web pages that improve accessibility. The rationale is that they produce the text-only page on-the-fly (hence the webmaster does not have to cope with the burden of maintaining redundant copies), and they can produce pages that are more accessible than the original ones (by dropping some of the contents, text transcoders remove also some of the existing defects).

While not being ruled out by accessibility guidelines (*e.g.* WCAG 1.0; Section 508) text transcoders have renewed discussions on the role and appropriateness of text-only pages, seen by many as second-level pages for second-level users. As a consequence, the adoption of text transcoders is often discouraged (*e.g.* the accessibility requirements issued by the Italian government [10] explicitly rules out text-only pages, regardless whether they are dynamic or not). However, so far, no studies exist on usability of the pages produced by text transcoders.

^aScientific advisor for the manufacturer of the transcoder used in the study described in this paper.

Several myths surround text-only pages. One myth is that they are already accessible (which is not true because of possible barriers due to tables, forms, unskippable navbars, badly chosen link labels); another myth is that text-only pages are a second, redundant web site (which is not true if one uses a transcoder, for example, as this is simply a mechanism that produces a new user interface for the same content); a third myth is that these pages are useless to people with no vision (which is false because content can be moved, reorganized, or added to improve user experience); and a final myth is that text-only pages are detrimental to the user experience of sighted users (which may be false because a simpler layout may lead to fewer navigation errors and to a better understanding).

Text transcoders are a technically viable solution when a straightforward translation of a web user interface is needed. For example, normal graphical pages can be transformed so that the visual layout, and in part the interaction structure, can better adapt to specific devices used by visitors (*e.g.* using screen readers or mobile devices). This happens because *user bandwidth* [19] can be increased by removing, rearranging or modifying the page contents or the interaction structure (*e.g.* by modifying the sequence of steps, decisions and actual actions that are needed to accomplish a goal) to make them more suitable to the sensory and motor capabilities of users. Therefore, as argued in [4], when appropriately installed, configured and customized, text transcoders have the potential of supporting a higher level of quality of use for certain users than the original web site does.

In fact, a text-only version of the web site can be an opportunity to customize web contents, navigation and presentation so that it can better suit people bound to low communication bandwidth (like slow modems), limited interaction and display tools (like PDAs or cell phones) or alternative channels (like screen-readers).

But because text transcoders drop part of the original content (*e.g.* images, Flash, applets, scripts), because they dramatically change the layout and structural HTML of pages (*e.g.* removal of layout tables), and because the customization that they support is somewhat limited, it is not yet clear whether resulting pages are beneficial at all. In fact, so much is changed and so many things have been removed, that it is likely that many user goals which could be reached in the original web site, would become unachievable. Thus, an important research question is: *what kind of impact has a text transcoder on users?*

The purpose of this paper is to describe an experiment aimed at determining the usability of transcoded pages with respect to disabled users. After a brief survey of known transcoders and a discussion of their most important features, the experimental methodology is illustrated, then the results obtained from the experiment are discussed, and conclusions are drawn.

2 A survey of text transcoders

Transcoders take as input a web page, process it by applying certain transformations, and return the transformed page and new HTTP headers to the user's browser. Very often the transformations implemented by transcoders can be split into two sets: those that are built-in and those that can be customized by the transcoder administrator (often called filters or annotations).

The following criteria are useful to classify existing transcoders:

- Whether they are server-dependent or totally independent from web servers. In the latter case transcoders have to follow the HTTP protocol to communicate with servers;

in the former case an *ad-hoc* protocol is used. While not affecting usability of transcoded pages, this aspect affects the applicability of a text transcoder: those based on HTTP are more loosely coupled to the web servers, and therefore can be more easily deployed.

- The objects upon which the transformations can be applied, including HTML contents, CSS external files, JavaScript external files, images, multimedia objects, HTTP headers. Few text transcoders process CSS; few of them are capable of limited processing of JavaScript; none process multimedia objects. Some of them process images (*e.g.* for detecting moving images and banners and for changing image resolution).
- Whether transformations change or not the page layout. In the former case they remove structural elements like frames, layout tables and content like images and video. In the latter case transcoders transform pages by adding invisible parts, like ALT text, style rules, JavaScript code, elements like NOFRAMES, NOSCRIPT, LABEL but the original layout is not affected. From a usability perspective, each approach has its pros and cons: improving the page layout makes it more suitable for users that maximize the windows or increase the text size. Preserving the original layout means to preserve also the original graphics and interactive features, and this in turn may help if these features are properly designed.
- If transformations are limited to a single page or if they apply to more than one page. For example, if a FRAMESET can be displayed only as separate transformed pages (one per frame), or if transformed frames can be simultaneously rendered in a single output page.
- If the transcoder handles correctly HTTPS, cookies, authentication, GETS and PUTS with all possible types of parameters. This affects directly effectiveness and security of users.
- If the transcoder is robust enough to handle badly written HTML, CSS or JavaScript code. Few HTML pages are based on valid code, and code invalidity may lead to assistive technology and browsers malfunctions, reducing the effectiveness of users.
- The robustness of the annotations: they need to be triggered by appropriate features within the source data (HTML, CSS, JavaScript, HTTP headers), they often need to extract some data, process it, to then produce required output. If the syntax of the source data changes then it is likely that some annotation is not triggered anymore, or if it is, that it produces the wrong output. Although this aspect does not affect usability of the transcoded pages, it is a factor that web masters need to assess when deciding to deploy a transcoder.

In [20], Mankoff et al. discuss the problem of web accessibility for people with severe sensory or motor disabilities. They present a number of approaches for making the web site accessible to these users (for example, by providing logical control through keyboard shortcuts, by wrapping in some modality part of the page so that limited human output signals can be overloaded with other meanings). They propose a prototype of a browser that is capable of being operated with two binary signals. They also propose a transcoder that adds large buttons for navigating within the page, that enlarges and highlights available links, that adds “Back” and “Forward” buttons to the page, and that adds the title attribute to the links in the page after pre-fetching destination pages and using their titles. The transcoder appears to

be a proof-of-concept prototype, as some of the functionalities are not completed. Feasibility and effectiveness of adopted techniques are also to be empirically evaluated. For example, adding the title attribute to links with information automatically gathered from destination pages is likely to improve the quality of labels, but it requires that the transcoder pre-fetches those pages, dramatically reducing its response time, which in turn affects negatively usability.

Mankoff and colleagues [16] survey client and server-side automatic transformation tools. Authors discuss the support that transformation tools can offer to disabled users with a very limited human bandwidth (*e.g.* people using *sip-and-puff* interfaces), and index the features of these systems by disability.

LIFT Text Transcoder (LTT) is the text transcoder we used in our study. The goal of LTT [25] is to produce high quality text-only pages, based on rewritten HTML code and a set of CSS files. LTT is server-independent, efficient (due to the reduced size of produced pages that lack all images, JavaScript, objects), fairly robust (it supports HTTPS, HTTP authentication, GETs, PUTs and some JavaScript static analysis), relatively easy to install, and customizable (by the manufacturer). Annotations are written in an XML/XSL proprietary language that can be used to change the content of the page being produced. Built-in transformations include simultaneous rendering of frames, liquid layout and linearization of layout tables (data tables are not linearized), re-sizable text, enlargement of link text; some options are selectable by the end-user, including switching between a few alternative color-themes and displaying links as if they were large buttons to the benefit of low vision or motor disabled users.

MUFFIN [3, 2] is a Java-based, server-independent HTTP proxy, that “. . . includes several filters which can remove cookies, kill GIF animations, remove advertisements, add/remove/modify arbitrary HTML tags (like blink), remove Java applets and JavaScript, user-agent spoofing, rewrite URLs” [3]. Its goals are security and privacy of the user of the browser, rather than accessibility. Its filters (written in Java) allow changing of arbitrary parts of the HTML code of the page, of the HTTP headers and even parts of images (*e.g.* removing or changing the looping behavior of GIFs). We believe it does not support generation of new CSS-based liquid layout and re-sizable text. MUFFIN supports HTTPS, and by default MUFFIN does not remove JavaScript, images and objects (which are therefore available for the end-user). In general (*i.e.* without properly written filters) MUFFIN does not yield web pages that are more accessible than the original ones.

`mod_accessibility` [18, 17] is a transcoder implemented as an output module for the Apache web server (hence being server-dependent). Its goal is similar to LTT: to improve the quality of output pages, and to make them more accessible than their original versions. However it does not necessarily strip-off images, JavaScript or multimedia contents. It offers the end-user a menu of different viewing options, one of which produces a text-only view of the page. Such a view includes simultaneous rendering of all the frames, linearization of the web page contents, generation of a liquid layout with re-sizable text controlled by proper CSS rules, and navigation bars moved to the bottom of pages. We believe this transcoder supports HTTPS, GETs and PUTs. By being an Apache module, and thanks to its caching strategy, `mod_accessibility` appears to be very efficient. However it does not support definition of new filters (unless one extends the Apache server by writing another output module that runs on top of `mod_accessibility`) and therefore it cannot be used to add site-specific contents (like new ALT attributes).

BETSIE [1] (BBC Education Text to Speech Internet Enhancer) is a Perl-based CGI program (server-independent) whose goal is to "...rearrange the content on the page in such a way as to make it more legible for people using text to speech converters or screen readers" [1]. BETSIE produces text-only pages whose content has been linearized (but it linearizes also data tables, reducing thus their accessibility), it produces pages featuring a liquid layout with re-sizable text, it strips-off images, JavaScript, Java applets but it keeps multimedia objects. It moves navigation bars that it can detect to the bottom of the rendered page, and it marks links that are external to the web site. It does not work with forms that require JavaScript. To the end-user it offers a number of selectable color-themes, selectable sizes and font families for text. BETSIE does not support explicit filters, but one can extend it by writing new Perl scripts. It appears that BETSIE is not able to handle correctly HTTPS or complex forms. The lack of explicit filters limits its customizability and flexibility of use, preventing addition of site-specific contents.

`textualise` [11] is a transcoder whose goal is to make the text-only pages more accessible. Like BETSIE, it offers a number of selectable color-themes, selectable sizes and font families for text. We believe `textualise` does not support explicit filters and no information is available regarding its architecture. The available demo shows that it moves some navigation bars to the bottom of the page and that it strips off all images, JavaScript and multimedia.

The transcoder described in [14] is a proxy server able to filter text, images, JavaScript and multimedia according to end-user preferences. The user can select the resolution with which images are to be rendered and whether JavaScript and other objects are to be sent to the browser. The transcoder seems to be able to translate HTML into WML. The *InfoPyramid* model [21], upon which it is based, defines the abstract transformations of information between different modalities (text, images, video, audio) and along several levels of detail (in terms of resolution, color depth, display size, streaming bit rate, compression format). We believe the transcoder can implement the basic HTML and CSS transformations that are supported by other transcoders we reviewed; it is not clear whether it can simultaneously render frames within an automatically generated liquid layout, and it is not clear whether explicit filters are supported and how, since its processing mechanism is based on a mathematical model that determines the level of detail that is appropriate for the target device.

Access Gateway [6, 7] is a proxy server aimed at producing web pages that can be customized by the end-user (supposedly a person with impaired vision) via choices from a very long list of options. It can simultaneously display frames, it is claimed to extract text from Flash objects, it can enable scripts and multimedia objects, it can perform a number of transformations on specific HTML tags, and it appears to support HTTPS, cookies and forms. It does not support new filters, however.

Accordion [9] is a special experimental browser designed for PDAs. It summarizes the page contents by identifying *semantic textual units* and allowing the end user to zoom on them, and between them if they are nested. Complexity of pages, and strategies to cope with it, are factors that may play a key role in accessibility of web pages. Even though Accordion has not been designed for accessibility, the ideas behind it could be used in a transcoder to help disabled users to access and use web pages. The paper presents an experiment aimed at detecting efficiency of users of this browser; unfortunately only informal results are given, which cannot be generalized at all. In addition they are limited to user efficiency.

AcceSS [22] is a transcoder aimed at simplifying and summarizing web pages for the benefit of disabled persons. Simplification is achieved through a decomposition and recombination process that aims at identifying page *landmarks* (like headers and footers) and *pagelets* (“self contained logical regions within a page that have a well defined topic or functionality”). This process is based on patterns that depend on the nature of pages, requiring therefore that pages be classified into a predefined set of genres (like e-commerce). Once a page has been deconstructed, AcceSS produces several new output pages: (i) one that contains the global navigation navbar (if any), (ii) a “guide dog” page that contains a description of the original page and a directory of its links, and (iii) a preview page, with no images or multimedia, for each main story contained in the original page. In this way a disabled user could explore the different parts of the original page in a more informed way, with many clues indicating where he/she is and where he/she can go.

Although the potential of AcceSS for improving web accessibility is high, at present there is very little experimental evidence for that. A limited and preliminary experimentation yielded some relatively high figures about the error rate of the classifier and of the decomposition/recomposition process, suggesting that some of the heuristics used in the process have to be tuned differently. Another informal and limited experiment was aimed at exploring usability of resulting pages (with two visually impaired users). Results are encouraging, even though the opportunities for users to get lost and not to find what they look for might be higher since there are more pages to navigate through than in the original web site. In addition, currently there is no data that allows any generalization of these results across different web sites and across different users disabilities, and no analysis is available on which usability factor (effectiveness, productivity, satisfaction) is affected more by AcceSS^b

Crunch [12] is another transcoder aimed at simplifying web pages. It focuses on removal of clutter (banners, logos, clusters of links), which is identified through a heuristic multi-pass process that can be tuned by the end user (*f.e.* by changing the ratio of link-text to non link-text). Also in this case a very limited experimentation has been performed so far, yielding no data that can be generalized in any way to determine its effectiveness or usability.

Other tools support a model-based deeper transformation of the user interface to suit different devices (*e.g.* based on retargeting or on plastic user interfaces [24, 23, 8]) rather than simply translating the more external layer of a user interface. But these tools are still confined to testing labs, and their analysis goes beyond the scope of this paper.

2.1 Transformations achieved by text transcoders

Transcoders, unlike gateways, apply a number of transformations to normal web pages and HTTP headers and return the transformed page and headers to the user’s browser. Transformations include (see figure 1): (i) built-in transformations that remove contents (*e.g.* images); (ii) built-in transformations that change the page structure and layout (*e.g.* producing a liquid layout); and (iii) customized transformations driven by annotations. As a consequence, usability of transcoded pages can be affected by any combination of these factors.

The transformations that were used in our study are representative of the ones that are usually implemented for accessibility purposes, which include:

- transcoded pages are stripped of all the original CSS and JavaScript code, and new CSS

^bFurther experiments are being planned for the short term (private communication).

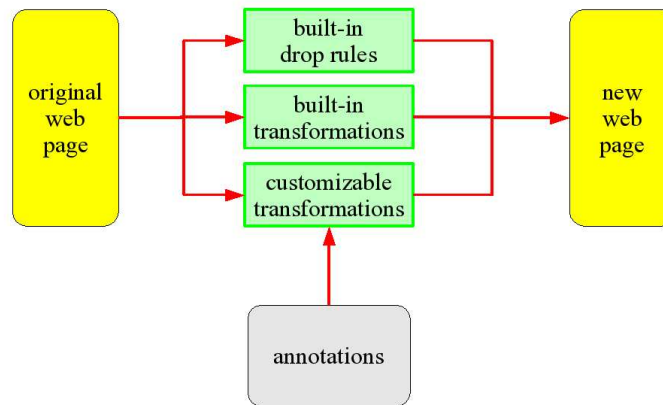


Fig. 1. The three mechanisms that affect transcoded pages

rules are added that yield a liquid design of the page and of the text;

- automatic redirects are replaced by server-side redirects;
- textual links that are too close each other are separated by white space or other characters (through annotations);
- scripts and event handlers are removed; when possible, scripts are replaced with content of NOSCRIPT clauses;
- frames and framesets are replaced by corresponding contents; framesets are linearized and frames are displayed simultaneously;
- layout tables are removed and their content is linearized, whereas data-tables are preserved; through annotations they can also be appropriately marked up;
- images are replaced by their ALT or TITLE; imagemaps are replaced by lists of links;
- new ALT for image buttons are added (through annotations);
- new TITLES for pages and frames are added (through annotations);
- navigation bars are reformatted to be displayed horizontally to better use the available screen space (through annotations);
- certain controls in the page (*e.g.* search box or navigation bars) are moved to its bottom (through annotations);
- page headings (H1 to H3) are added (through annotations); in certain cases they replace images;
- forms are linearized and labels are properly positioned; form control titles are also used when appropriate (through annotations);

- objects and applets are replaced with their textual equivalent if any; they are stripped otherwise.

3 Research methodology

3.1 Research goals

The purpose of the study is to determine usability with respect to disabled users^c of transcoded pages obtained through a customized text transcoder applied to a web site that is not accessible.

We framed this question into a comparative experiment aimed at measuring usability under two experimental conditions: use of the original web site *vs.* use of the web site through the customized text transcoder. We wanted to test whether

- dynamically generated text-only pages can be more accessible than the original ones and therefore if the increased accessibility level is due to the built-in and customized transformations;
- the increased accessibility can balance the removal from the original pages of certain features (careful graphical layout and non-textual contents).

In this way we can test the actual usability of transcoded pages with respect to disabled end-users in the typical situations in which dynamic text-only pages are deployed, *i.e.* as an alternative user interface for an existing web site that is not accessible. And determine if these dynamically created text-only web sites are a viable solution.

We decided to run these experiments on web sites that are not accessible: if they were, then it is likely that no positive difference in usability would have emerged from the experiments, unless the annotations were affecting other aspects of the web sites besides accessibility (like its information architecture). But we did not want the transcoders to change these aspects of the tested web sites.

We explicitly do not consider the following factors that affect the overall usability and quality of transcoders: how easy it is to write and test filters/annotations, how robust annotations are, how robust the transcoders are themselves, if they can translate into other languages than HTML and CSS, and how well they support HTTPS, cookies, GETs and PUTs. These are important factors that contribute to the usefulness (and thus global quality of use) of a transcoder, but go beyond the scope of this paper because they do not affect directly usability with respect to end-users.

3.2 Preliminary study

To better understand the problem, we ran a preliminary experiment. We involved 11 students, professors, and staff whose age ranged from 16 to about 50: 3 participants were affected by low-vision and 8 were blind. These persons were asked to carry out two tasks on an e-commerce web site, one with the transcoder and one without. They used whatever assistive technology they were used to, and in a physical setting that they were familiar with. The original purpose of this first study was to gather qualitative and quantitative data about

^cWe adopted the ISO definition of accessibility: “*usability of a product, service, environment or facility by people with the widest range of capabilities*”[15] and focused thus on users effectiveness, productivity and satisfaction.

users preferences and performance. However it was quickly discovered that several factors were hindering collection of proper data, namely:

1. A large variability of user skills in using screen-readers (some users were literally flying over pages at a pace that made it difficult for the sighted experimenters to follow their actions; others were moving across pages by using only few keystrokes (left arrow, right arrow, enter and back) making the whole experience very tiring and error-prone).
2. A large variability of user skills in dealing with features of web pages. For example, due to the way the interaction structure was layed out, several users inadvertently bought the same article 2 or more times. None bothered to check the number of items shown in the shopping cart when checking out, because it was rendered very poorly.
3. Little knowledge about accessibility features. For example no user took advantage of the *skip to page contents* link that would allow him/her to jump over long navigation bars.
4. Minor differences in the way in which links were named made big differences in the success rate. For example a user learned that in one page the name of a band was “Rolling Stones”, and was unable to locate the correct link in another page because the link was spelled as “The Rolling Stones” (the participant was using the *Link list* feature of the screen-reader and he typed only “r”).

3.3 Customization of the transcoder

On the basis of what we learned in the first user testing experiment, we conducted a larger test (for practical reasons) on a different web site. The test site belongs to an local Italian government agency^d and is not accessible: pages lack textual alternatives for iconic buttons, are poor in terms of liquid layout and re-sizable text, some areas feature a low foreground/background contrast, forms controls are not appropriately labeled, and no intra-page navigation is supported (*e.g.* sectioning, *skip-links* links, links with non-unique text labels).

Annotations for LTT were constrained to process existing content of web pages and smooth some of the accessibility barriers, not to add new content or to significantly alter the information architecture of the web site. The only new content was page and frame titles, page headings, ALT for iconic buttons, hidden *skip-links* links, access keys for global navigation links, and a small table of contents on each page.

In this way the original content and information architecture of the web site was not affected by annotations and the transcoder was tested in a typical deployment situation, where a web master is expected to adapt the transcoder to the specific coding conventions of the site and fix most common accessibility barriers. Other than following generic accessibility requirements, pages were not customized to fit particular needs. Figure 2 shows a fragment of a transcoded page of the test site (seen through a browser running on a PC)^e. The effect of some annotations can be noticed on the horizontal layout of navigation bars, in the initial table of contents with accesskeys, in the sectioning of the page contents (bold text is coded

^dwww.regione.fvg.it

^eThe annotated web pages can be seen at transcoder.usablenet.com/tt/http://www.regione.fvg.it.

with HTML headings H1, H2, H3). For example, through a site-wide annotation all the links appearing in navigation bar on the left (with a dark background) are formatted into the 3 lines below the heading “Istituzioni regionali”; then under “Attività in regione” are the links and subheadings of the (multicolored) right navigation bar; and similarly for the remaining content of the original page.

We used LTT because of its flexibility and availability to us. It was installed on servers that were physically located in UK and therefore on a remote location compared to participants (university labs and other places in Italy) and to the original web server (a different town in Italy).

Annotations for LTT are XML fragments, stored in files that are separate from the original web pages and owned by the administrator of LTT. Each annotation refers to an element type (*i.e.* a tag name) of the DOM of the document; it can be page-specific (*i.e.* restricted to a single URL) or site-wide, it has a match condition specified through an XPath expression, and a transformation section. The target element is the first element of the DOM that matches both the specified element type and the condition. The transformation section says how to transform target elements. The input page is parsed, its DOM is generated, built-in transformation rules are executed, the transformation sections of applicable annotations are applied to target elements, the DOM is modified, it is pruned by applying dropping rules and finally the resulting HTML page is generated. This process is based on a pipeline architecture that creates and applies several XSL style sheets.

For the test site we developed about 140 annotations, some of which are shown in figure 3.

3.4 *Experimental plan*

The main experiment involved 29 participants (8 females and 21 males): 17 were blind, 7 had low-vision and 5 had motor disabilities. Nineteen used a screen-reader (with or without a Braille reader), 4 used screen magnifiers, 1 used modified mouse and keyboard, and the remaining ones did not use any specific assistive technology. The age ranged from 21 to 58, with a mean of 36.6 (figures 4 and 5). As shown in figure 5, half of the participants use the PC for at least 30 hours/week, 25% use it for less than 12 hours/week; as for the Internet usage, half of them use it for at least 10 hours/week, 25% for no more than 3 hours/week.

As a screening criterion we required only that all participants have prior experience with computers and with the Internet.

Even though our sample is not randomly chosen and we did not finely control the participants level of experience (in using the PC, the assistive technology, the Internet, the web site being tested), it turns out that the sample is nevertheless quite homogeneous. In fact 50% of the participants use the PC between 12 and 40 hours/week, and 50% of them use it for more than 30 hours/week. And 50% of them use the Internet for more than 10 hours/week. When we restrict the analysis to participants with no vision, the ranges are even smaller: 50% of them use the PC between 20 and 40 hours/week, and 50% use it for more than 56 hours/week. We expect therefore that there is less variability in the experience level than what was noticed in section 3.2.

Each participant was asked to carry out (in random order and in a familiar setting with any assistive technology he/she was used to) 5 information finding tasks, two with LTT, two without and one with another transcoder (Access Gateway). The latter task was needed in



Fig. 2. Screenshot of the original page www.regione.fvg.it/welcome.asp (top) and the text-only version produced by the transcoder (bottom).

```

<annotation for="a" id="add_content_english_version">
  <description>Replace the content of the link English Version</description>
  <match>contains(@href, 'www.regione.fvg.it/inglese')</match>
  <execute> <set-content>
    English Version
  </set-content> </execute>
</annotation>

<annotation for="img" id="add_alt_documentazione">
  <description>Replace img documentazione</description>
  <match>@src='img/documentazione.gif'</match>
  <execute>
    <replace-element name="h3" keep-content="no"/>
    <set-content>Documentazione</set-content>
  </execute>
</annotation>

<annotation-map>
  ...
  <page level="1">
    <file>industria.xml</file>
    <path>/industria.htm</path>
  </page>
  ...
</annotation-map>

```

Fig. 3. Examples of annotations: the first one is a site-wide annotation that replaces, on every visited page, the content of the A element that has the given HREF attribute with the new text. The second annotation, that according to the annotation map applies only to the page whose URL ends with /industria.htm, applies to the IMG element that has the given SRC attribute. Such an element is replaced with a new H3 element whose content is “Documentazione”.

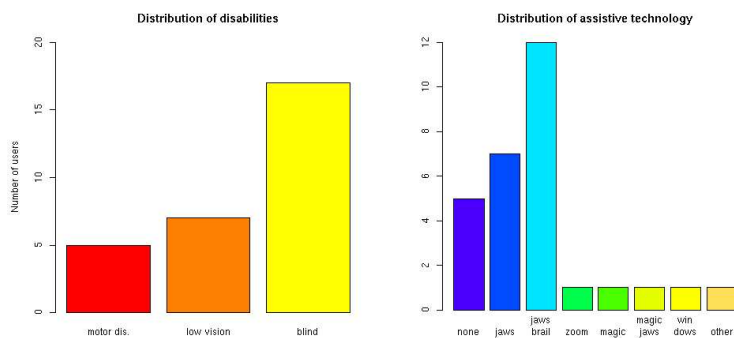


Fig. 4. Distribution of participants according to type of disability and of assistive technology

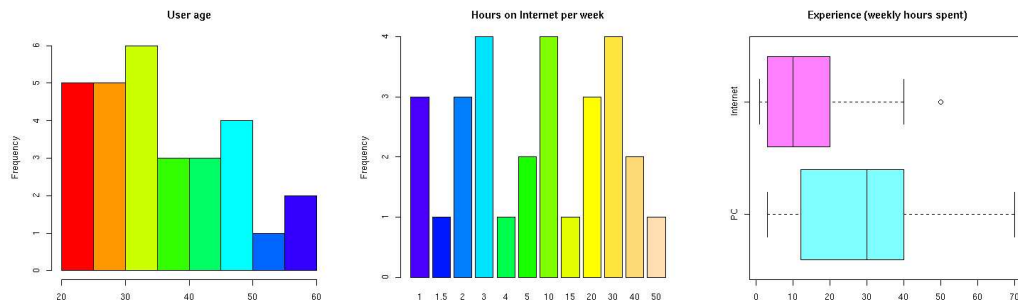


Fig. 5. Distribution of participants according to age (left), of weekly hours spent using the Internet (center) and boxplots of the weekly hours of use of PC and Internet (right). In the boxplots, whiskers represent the minimum and maximum values, dots are outliers, and rectangles represent the 3 quartiles.

order to reduce the test effect by simulating a single blind test: because subjects were not sure of which was the actual transcoder we were experimenting with, this limited their bias in their opinions. The outcomes of this task were not used to draw any conclusion.

We split the four main tasks into two pairs: one pair of tasks was simpler than the others, and required subjects to browse 2 or 3 pages to find the required information; the other tasks required in addition to fill-in a form (see figure 6 for details).

In order to balance the learning effect, we randomized both the difficulty level of tasks and the treatment (with/without LTT). The task with Access Gateway was randomized between the 4 main tasks.

task	level	short description
A	simple	find postal address and telephone number of <i>Ufficio Regionale della Protezione Civile</i>
B	simple	find the URL address of <i>Agenzia Regionale della Sanità</i>
C	complex	find job opportunities in administrative fields applied to health management
D	complex	find addresses of farm holidays in the area of Bertolo

Fig. 6. The four main tasks (the actual formulation was in Italian, which included a more detailed 5 lines scenario description).

All the work sessions were videotaped and a post-task questionnaire was submitted to the subjects. The purpose of the questionnaire was to elicit information about satisfaction, perception of effectiveness and of productivity. Most answers were framed as 5-point Likert scales. Although polarity of scales of questions was sometimes reversed, questionnaires were carefully read to the subjects and were actually filled in by the facilitators, to avoid mistakes and further frustration.

The independent variables included whether LTT was used during execution of a task or not, and the type of tasks (easy *vs.* complex). The dependent variables characterizing usability included subjective ones (*e.g.* opinions asked to the participant, like how easily the information was found) and performance-related ones (*e.g.* time to complete a task, level of completion). The dependent variables were associated to the basic usability factors (effectiveness, productivity and satisfaction) as shown in figure 7, which presents also the statistically significant results.

4 Results

4.1 Analysis of results

Visual inspection suggests that data is not normally distributed: no data is symmetric, features a bell-shape and agrees with the normal probability plot. Therefore the null hypotheses (that LTT does not affect a certain parameter) has to be evaluated with non parametric statistical tests, like the Wilcoxon signed rank test for paired samples. This is a relatively weak statistical test, and therefore the figures we present are conservative; in addition only differences featuring a significance value (*i.e.* probability of being wrong) of 5% or less are shown.

Figure 7 shows the data produced by the experiment, along with significance figures where appropriate. Non significant variables were omitted. No statistically significant difference in favor of the “no LTT” condition was found.

Figure 8 shows results obtained when restricting the analysis to blind users, users with sight deficiencies or users that are not blind^f. Also in such a case, no significant difference in favor of the “no LTT” condition was found except for a single variable (namely the time required to complete a task for sighted users). Even though LTT improved some variable (*e.g.* Q_4 for no vision and for visually impaired persons), it appears that those improvements are confined to persons with visual disabilities. LTT improved some variables (*e.g.* Q_4 , Q_5 , Q_7) for those that can see.

Strangely enough, we were not able to identify significant correlations between any of the dependent variables of interest and the level of experience in using the Internet. For no variable the Spearman correlation index ρ^g of the variable with the level of experience in Internet (in magnitude) was greater than 0.35. This suggests that the changes induced by LTT lead to differences that are independent from user experience and suit both the novice and the expert users.

Figure 9 shows correlation indexes for some of the variables. The variables Q_1 , Q_7 , CL, GU, NP, T, NPS and NE agree also for the difference of the values obtained under the two experimental conditions (*i.e.* the studied effect). Therefore we believe that the questionnaire that was used is reliable.

Given the data just presented, the following conclusions can be drawn on the three usability factors of interest.

Effectiveness

- LTT improved Q_1 (“I found the required information”) and Q_2 (“I was tempted to go elsewhere”), for simple, complex and both tasks; for example, the mean score for Q_1 on simple tasks moves from 3.5 (between *mildly disagree* and *cannot say*) to 2.2 (close to *mildly agree*). The difference is even stronger for users with visual disabilities.

^fNo results are shown for users that are not visually impaired as the sample is too small (5 users) to lead to differences that are significant.

^gWe report Spearman’s ρ when significance level $p < 0.05$ and for the strongest correlations only. ρ measures the degree of correlation between the ranks of values of two variables; absolute values of ρ greater than 0.7 suggest strong correlations, and greater than 0.5 suggest weak ones.

- LTT did not improve Q_3 (“I always knew where I was in the site”), for which no significant difference was found.
- LTT improved Q_4 (“I’m satisfied with the results”) for simple tasks and overall, but not for the complex ones, for which no significant difference was found. For blind persons, the difference in favor of LTT is greater; on the other hand, for people that can see, the difference is smaller.
- LTT improved also the difficulty ranking of tasks Q_7 ; when using LTT the mean difficulty rank is for simple tasks 3.24 (1 being the most difficult, and 4 the least), while without LTT it is 2.38. Overall the means are 2.17 *vs.* 2.81. No significant difference was found on simple tasks. A stronger effect of LTT can be seen when restricting to particular categories of users.
- For more than 70% of the users, LTT would be the preferred choice for the next visit to the web site (Q_8). This is even higher (79%) for people with some visual disability.
- In terms of CL (task completion level), more than half of the participants that used LTT reached the 100% success level, *i.e.* they completely solved the problem they were asked to (for complex tasks, and overall; no difference was found for simple tasks), whereas without LTT the median is 0.5, *i.e.* half of them reached only half of the solution. Also the means show a marked improvement of about 50%.
- Similarly for PR, the proportion of tasks who exceeded a 50% success level. With LTT the proportion more than doubles for simple tasks; it shows no difference for complex tasks; and overall it changes from 65% to 79%. For no vision users, it changes from 32% to 62% (almost doubled), and from 39% to 60% for users with visual disabilities.
- GU, the proportion of users that gave up, changes for complex tasks from 51% to 24% when using LTT; no difference was found overall nor for simple tasks. It dropped from 52% to 20% for no vision people, and from 48% to 19% for those with visual disabilities.
- NE, the number of wrongly visited pages, is smaller when using LTT and changes from 2.59 to 0.97 (for simple tasks) and 2.67 to 1.57 (overall); no difference was found for complex tasks alone. No difference was found either for the number of errors per hour. A similar decrease can be noticed also when restricting to no vision people and to those with visual disabilities.
- NB, the number of activations of the back button, is smaller when using LTT on simple tasks (it changes from 1.93 to 0.59) and overall (1.95 to 1.40); no difference was found for complex tasks. No significant difference is shown for any user category.

Productivity

- NP, the mean number of visited pages, when using LTT changes from 4.59 to 2.93 for simple tasks, and from 5.05 to 4.21 overall; no difference was found for complex tasks alone. No significant difference is shown for any user category.
- T, the mean absolute time to complete a task, decreases from 465 to 301 seconds for simple tasks; no difference found for complex tasks or overall. Interestingly, for no vision users LTT reduced the mean time from 545 to 337 sec whereas for sighted users LTT *increased* the time from 438 to 600 sec.

- For the mean time to successfully complete a task, no difference was found.
- NPS, the mean number of visited pages given a full success level, changes from 8.14 to 5.63 for complex tasks, and from 5.56 to 3.90 overall; no difference was found for simple tasks.
- The mean subjective assessment of effort, Q_5 , changes from 2.5 (between *quite high* and *cannot say*) to 4 (*quite low*). Thus LTT improves it, across all user categories.
- No significant difference was observed on relative metrics of productivity, like completion time given a certain success level, number of visited pages given a success level or number of errors given a success level.

Satisfaction

- As commented above, LTT generally improves Q_1 , Q_2 , Q_4 , Q_5 and Q_8 .
- LTT improves Q_6 , the evaluation of presentation quality; the mean changes from 3.5 to 5.4 (in a scale 1 to 7, where 1 is *very bad*).

Informal results Participants and experimenters opinions, gathered during the execution of the tests, include:

- Many participants appreciated the ability of transcoded pages to be enlarged and to resize the text. A drawback was that browser and window features (like the BACK button or the mouse pointer) could not be enlarged.
- Many users appreciated the ability to display links as if they were buttons, simplifying their usage. Some users were pleased by the ability to choose a different color scheme for the background and the foreground.
- Even though transcoded pages had the *skip-links* links, almost no user activated them; but most of them complained about the repetition of the same links over and over. This partly confirms that the capabilities of our sample of participants were quite uniform.
- Even though the linearization of the page content has often helped users in locating the needed information, they *forgot* to scroll it vertically, with the result that large portions of pages were rarely seen.
- Expert users of screen readers were the least affected by the accessibility barriers shown by the original web site. Therefore, for them, the benefit of LTT was minimal.
- Users of outdated versions of screen readers (*e.g.* JAWS 3.5) were unable to appropriately use frames in the original web site. Therefore LTT enabled them to complete the tasks.
- Most of the participants appreciated the new content added with annotations: ALT for link images and new headings, LABEL/FOR for form controls. One benefit of new headings was that they helped many users to quickly recover from navigation errors.

4.2 Discussion

The effects of LTT are not completely uniform (*i.e.* when no difference can be proved, this occurs mostly for the complex tasks and for the minority of users with non visual disabilities),

and this suggests that LTT, and the accessibility changes that it applies to a web site, cannot improve usability for tasks that are beyond a certain level of difficulty, or for which the interaction design is too poor, or where the presentation is graphically very poor.

Q_3 (“I always knew where I was in the site”) shows no significant difference (medians are at 2.1 and 1.8, for the “no LTT” and LTT conditions respectively, in a scale 1–5 where 1 is better), and this can be explained by the relatively poor information architecture (logical structure, navigation systems and labels) incorporated into the original web site. LTT annotations did not affect it as the original information architecture was included in transcoded pages.

Though 12 users (out of 29) could exploit in some way the graphical pages, the relatively poor graphical layout of transcoded pages had only limited consequences on the usability scores discussed above. However, since most of the differences are strongest for users with some visual disabilities, it is likely that those users take most of the advantages from LTT. Interestingly, LTT slowed down users that have some visual capability. A tentative explanation is that the poverty of the text-only layout, and the additional scrolling that was necessary, required more time. But it also suggests that users were more motivated to continue using the web site before giving up.

We couldn’t detect any usability difference due to the addition of access keys, of *skip-links* links and table of contents, that were very seldom used.

In general, subjective data shows a more marked effect than objective variables, for which in many cases no significant difference was found. We believe that by choosing more complex tasks this difference could be made more explicit.

We believe that the differences shown by this experiment apply to a larger population than just the 29 users of our sample, provided that such a population shares the same characteristics (described in section 3.4). An important condition to consider is that our sample consisted of 58% blind persons, 24% persons with low-vision and 17% persons with motor disabilities. It may be the case that for visually unimpaired persons the effect of text transcoders could be different, as indeed some of the data suggest. In particular our sample of users under-represented users with motor disabilities and did not include any deaf, hard of hearing users nor any user with cognitive disabilities.

The differences we were able to show should apply to most information web sites that require browsing, and for tasks similar to the ones we used in the experiment: fact finding with possibly few simple forms to be filled in.

5 Conclusions

This experiment demonstrates that dynamically created text-only web pages do not decrease usability of a web site when used by disabled people that share the same characteristics of the sample we studied, described in section 3.4, but on the contrary, a text transcoder configured to fix the most common accessibility barriers improves users effectiveness, productivity and satisfaction despite the removal of content and the automatically generated page layout. This experiment shows that most of the effect is to the benefit of users that are visually impaired, but too little experimental evidence is available at the moment to rule out benefits to other kinds of users.

Although the same increase in usability is likely to be achievable by retrofitting accessibility to the original web site, this result is important because (i) text transcoders can be applied

to inaccessible web sites to offer a *temporary* accessible user interface, and (ii) they can be applied to offer an *alternative* accessible user interface that better suits these kinds of users. In fact we believe that any specific adaptation of transcoded pages to the needs of disabled users (*e.g.* by suitable use of colored sections and icons, by arranging the page contents differently) is likely to dramatically improve usability also for users whose characteristics differ from those described in section 3.4, according to the claims in [4].

We believe the key features of LTT that support these findings are:

- the ability to simultaneously render all the frames of a framed page;
- the adoption of a liquid layout, fully controlled by CSS, with arbitrarily re-sizable text;
- the ability to customize the transcoder (via annotations or filters) for laying out navigation bars horizontally, adding ALT text to image buttons and images in general, adding section headings, and for labeling form controls.

Therefore we would expect to find similar results also with other text transcoders or CMS, provided that they support these kinds of transformations.

Additional investigations on how text transcoders affect usability of pages with respect to users with other characteristics are needed. A currently ongoing experiment aims at determining how usability of transcoded pages changes with respect to non disabled users. We plan to determine whether there are differences, in which direction and to what extent, and to use such data as a comparison baseline for measuring the increase in usability we reported in this paper. Another experiment on usability of transcoders with respect to PDA users has been partly described in [5].

Acknowledgments

Many thanks to Daniela Nicoli and Mery Pignatelli who helped us producing and collecting the data and to Luciano Picone for his administrative help. We thank also Enrico Scoda and Francesca Leon for their assistance in developing annotations. We are very grateful to all the participants whose time and effort were essential for this study. Many thanks also to Shadi Abu-Zahra for his focused early comments. This work was partially funded by a grant by the Provincia di Udine.

References

1. BBC Education. Betsie Home Page. www.bbc.co.uk/education/betsie. Visited Nov. 2004.
2. M. R. Boyns. The Design and Implementation of a World Wide Web Filtering System. muffin.doit.org/thesis/slides/mgp00001.html. Visited Nov. 2004.
3. M. R. Boyns. MUFFIN World Wide Web Filtering System: Documentations. muffin.doit.org. Visited Nov. 2004.
4. G. Brajnik. Achieving universal web access through specialized user interfaces. In C. Stary and C. Stephanidis, editors, *Lecture Notes in Computer Science Proc. of the 8th ERCIM UI4ALL Workshop*, Vienna, June 2004. Springer Verlag. www.dimi.uniud.it/giorgio/publications.html#specui.
5. G. Brajnik, D. Cancila, D. Nicoli, and M. Pignatelli. Do dynamic text-only web pages improve usability for pda users? In *First Usability Symposium Austrian Computer Society*, Vienna, Austria, Nov. 2005. www.dimi.uniud.it/giorgio/publications.html#vienna05.

6. S. Brown and P. Robinson. A world wide web mediator for users with low vision. In *CHI 2001 Workshop n. 14*, 2001. www.ics.forth.gr/proj/at-hci/chi2001/files/brown.pdf.
7. S. S. Brown. About the web access gateway. www.cl.cam.ac.uk/~ssb22/access.html, 2004. Visited Nov. 2004.
8. L. Buillon and J. Vanderdonckt. Retargeting web pages on other computing platforms with vaquita. In van Deursen and A. Burd, editors, *Proc. of IEEE Working Conf. on Reverse Engineering WCRE' 2002*, pages 339–348, Richmond, Oct 2002. IEEE Computer Society Press.
9. O. Buyukkokten, H. Garcia-Molina, and A. Paepcke. Accordion summarization for end-game browsing on PDAs and cellular phones. In *Human-Computer Interaction Conference 2001, CHI 2001*, 2001.
10. CNIPA. Requisiti tecnici e i diversi livelli per l'accessibilità agli strumenti informatici. www.pubbliaccesso.gov.it/normative/DM080705.htm, July 2005.
11. codix.net. Web accessibility: textualise. aquinas.venus.co.uk/solutions/products/textualise/index.html. Visited Nov. 2004.
12. S. Gupta and G. Kaiser. Extracting content from accessible web pages. In Harper et al. [13], pages 26–30.
13. S. Harper, Y. Yesilada, and C. Goble, editors. *Int. Cross-Disciplinary Workshop on Web Accessibility, W4A*, Chiba, Japan, April 2005. ACM.
14. IBM Research. IBM transcoding technology. www.research.ibm.com/networked_data_systems/transcoding. Visited Nov. 2004.
15. ISO. Ergonomics of human-system interaction — guidance on accessibility for human-computer interfaces. ISO/TS 16071. Technical report, International Standards Organization, 2003. www.iso.ch.
16. M. Ivory, J. Mankoff, and A. Le. Using automated tools to improve web site usage by users with diverse abilities. *IT & Society*, 1(3):195–236, Winter 2003. www.stanford.edu/group/siqss/itandsociety/v01i03/v01i03a11.pdf.
17. N. Kew. WebThing home page. apache.webthing.com/mod_accessibility. Visited Nov. 2004.
18. N. Kew. mod_accessibility for Apache 2. www.miswebdesign.com/resources/articles/mod_accessibility.html, 2003.
19. S. MacKenzie. *Fitts' law as a performance model in human-computer interaction*. PhD thesis, University of Toronto, Ontario, Canada, 1991. www.york.ca/mack.
20. J. Mankoff, A. Dey, M. Moore, and U. Batra. Web accessibility for low bandwidth input. In *Proc. of the Fifth ACM SIGCAPH Conference on Assistive Technologies*, July 2002.
21. R. Mohan, J. Smith, and C.-S. Li. Adapting multimedia internet content for universal access. *IEEE Transactions on Multimedia*, pages 104–114, March 1999.
22. B. Parmanto, R. Ferrydiansyah, A. Saptono, L. Song, I. Sugiantara, and S. Hackett. Access: accessibility through simplification and summarization. In Harper et al. [13], pages 18–25.
23. F. Paternò and C. Santoro. One model, many interfaces. In *Proceedings Fourth International Conference on Computer-Aided Design of User Interfaces*, pages 143–154. Kluwer Academic Publishers, May 2002.
24. D. Thevenin and J. Coutaz. Plasticity of user interfaces: framework and research agenda. In S. A. and J. C., editors, *Proceedings of Interact '99*, pages 110–117, Edinburgh, UK, 1999.
25. UsableNet Inc. LIFT Text Transcoder. www.usablenet.com/products_services/text_transcoder/text_transcoder.html, Feb 2005. Demo version available at demott.usablenet.com:8080/tt.

variable and factors	task type	NO LTT mean (sd)	LTT mean (sd)	NO LTT	LTT	p
Q ₁ I easily found the required information (median, 1-5, 1 is better) (E,S)	simple	3.5 (1.68)	2.2 (1.45)	4	2	< 0.001
	complex	3.8 (1.68)	3.1 (1.45)	4	2	< 0.048
	overall	3.69 (1.61)	2.64 (1.57)	5	2	< 0.001
Q ₂ I was tempted to go elsewhere to find the answer (median, 1-5, 5 is better) (E,S)	simple	2.6 (1.84)	3.6 (1.88)	1	5	< 0.033
	complex	2.2 (1.65)	3.4 (1.80)	1	5	< 0.003
	overall	2.40 (1.75)	3.5 (1.83)	1	5	< 0.002
Q ₄ I'm satisfied with the solution I found (median, 1-5, 1 is better) (E,S)	simple	3.1 (1.88)	2.0 (1.37)	3	1	< 0.005
	overall	3.2 (1.81)	2.3 (1.68)	4	2	< 0.002
Q ₅ Assess the effort required (median, 1-5, 5 is better) (P,S)	overall	2.5 (1.32)	4 (1.03)	2	4	< 0.001
Q ₆ Evaluate the presentation quality (median, 1-7, 7 is better) (S)	overall	3.5 (1.4)	5.4 (1.18)	3	6	< 0.001
Q ₇ Rank the tasks by difficulty (median, 1-4, 4 is better) (E)	simple	2.38 (1.11)	3.24 (1.02)	3	4	< 0.008
	overall	2.17 (0.55)	2.81 (0.54)	2	3	< 0.004
Q ₈ Proportion of users that would choose LTT/NOLTT for a next visit (E,S)	overall			0.28	0.72	< 0.001
CL Completion level (E)	simple	0.48 (0.41)	0.74 (0.41)	0.5	1	< 0.003
	overall	0.46 (0.40)	0.65 (0.42)	0.5	1	< 0.002
PR Proportion of tasks when completion level > 50% (E)	simple			0.31	0.68	< 0.005
	overall			0.65	0.79	< 0.013
GU Proportion of tasks when completion reason = subject gave up (E)	complex			0.51	0.24	< 0.030
NP Number of visited pages (P)	simple	4.59 (2.90)	2.93 (0.92)	4	3	< 0.004
	overall	5.05 (2.97)	4.21 (2.17)	4	4	< 0.039
T Task completion time in sec. (P)	simple	465 (355)	301 (274)	393	178	< 0.047
NPS Number of visited pages when success=100% (P)	complex	8.14 (3.43)	5.63 (1.96)	7	5	< 0.041
	overall	5.56 (3.34)	3.90 (1.83)	4.5	3	< 0.042
NE Number of wrongly visited pages (E)	simple	2.59 (3.22)	0.97 (1.59)	1	0	< 0.023
	overall	2.67 (2.82)	1.57 (2.04)	2	1	< 0.014
NB Number of clicks on back (E)	simple	1.93 (2.04)	0.59 (0.83)	1	0	< 0.012
	overall	1.95 (3.16)	1.40 (3.09)	1	0	< 0.045

Fig. 7. Dependent variables, their associated usability factor (Q_i means question, E, P and S stand for effectiveness, productivity and satisfaction). Results are given in terms of means, standard deviations, medians and significance values. *Type* represents the type of tasks being considered: simple, complex or both. For questions $Q_1 \dots Q_4$ the answer scale is: 1=fully agree, 2=mildly agree, 3=cannot say, 4=mildly disagree, 5=fully disagree; for Q_5 : 1=high ... 5=low; for Q_6 : 1=very bad ... 7=very good; for Q_7 : 1=difficult ... 4=easy; for CL: 0, 0.2, 0.4, 0.6, 0.8, 1

	variable	type of user	NO LTT	LTT	p
Q ₁	<i>I easily found the required information (median, 1-5, 1 is better)</i>	no vision	5	2	< 0.001
		vision impaired	5	2	< 0.001
Q ₂	<i>I was tempted to go elsewhere to find the answer (median, 1-5, 5 is better)</i>	no vision	2	5	< 0.001
		vision impaired	1	5	< 0.002
Q ₄	<i>I'm satisfied with the solution I found (median, 1-5, 1 is better)</i>	no vision	5	2	< 0.004
		w/ vision	2.5	1	< 0.010
		vision impaired	4.5	2	< 0.005
Q ₅	<i>Assess the effort required (median, 1-5, 5 is better)</i>	no vision	2	4	< 0.001
		w/ vision	2.5	4	< 0.036
		vision impaired	2	4	< 0.001
Q ₆	<i>Evaluate the presentation quality (median, 1-7, 7 is better)</i>	no vision	3	6	< 0.001
		vision impaired	3	6	< 0.001
Q ₇	<i>Rank the tasks by difficulty (median, 1-4, 4 is better)</i>	no vision	1	2.7	< 0.003
		w/ vision	1	2.7	< 0.002
		vision impaired	1	2.7	< 0.007
Q ₈	<i>Proportion of users that would choose LTT/NOLTT for a next visit</i>	no vision	0.30	0.70	< 0.014
		vision impaired	0.21	0.79	< 0.008
CL	Completion level	no vision	0.4	1.0	< 0.005
		vision impaired	0.5	1.0	< 0.002
PR	Proportion of tasks completed at level > 50%	no vision	0.32	0.62	< 0.015
		vision impaired	0.39	0.60	< 0.034
GU	Proportion of tasks with completion reason = subject gave up	no vision	0.52	0.20	< 0.006
		vision impaired	0.48	0.19	< 0.003
T	Task completion time in sec	no vision	545	337	< 0.015
		w/ vision	438	600	< 0.030
NE	Number of wrongly visited pages	no vision	2.7	1.76	< 0.011
		vision impaired	2.6	1.58	< 0.020

Fig. 8. For variables where a significant difference has been found overall (on all users and on both types of tasks), this table reports results obtained restricting the analysis to users with no vision (17 of them), with some sight capabilities (12 of them) and with impaired vision (24 of them), *i.e.* no and low vision ones.

param	simple	complex	param	simple	complex
Q ₂ and Q ₁	-0.75	-0.60	Q ₁ and Q ₄	0.70	0.73
NP and NE	0.72	0.69	Q ₁ and NE	0.70	0.34
CL and Q ₄	-0.56	-0.44	CL and NE	-0.48	-0.22

Fig. 9. Some pairs of variables are considered, and for each of them the values of Spearman's ρ are given, respectively for simple and for complex tasks. Negative correlations are due to the reversed polarity of the measurement scales.