Do text transcoders improve usability for disabled users?

Giorgio Brajnik[*], Daniela Cancila, Daniela Nicoli, Mery Pignatelli
Dip. di Matematica e Informatica
Università di Udine — Udine, Italy
giorgio@dimi.uniud.it

ABSTRACT

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 been scientifically studied yet.

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 wider population.

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 be deployed to automatically yield text–only version 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 current draft of the accessibility requirements issued by the Italian government [8] explicitly rules out text–only pages, regardless whether they are dynamic or not). However, so far, limited studies exist on usability of text transcoders, and more specifically on usability of the pages they produce.

Text transcoders are a technically viable solution when translation of a web user interface is needed. For example, normal graphical pages can be transformed so that the visual layout, and possibly 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* [15] 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). 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 have been reached in the original web site, would become unachievable.

Thus, an important empirical 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

[*]Scientific advisor for the manufacturer of the transcoder used in the study described in this paper.

W4A at WWW2005 10th May 2005, Chiba, Japan

Transcoders are proxy–like systems that take as input a normal web page, process it by applying certain transformations, and return the transformed page and 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.
- The objects upon which the transformations (built–in or customized ones) can be applied, including HTML contents, CSS external files, JavaScript external files, images, multimedia objects, HTTP headers.
- Whether transformations change or not the page layout, by removing structural elements like frames, layout tables and content like images and video. In the latter case transcoders transform the pages by adding invisible parts, like ALT text, style rules, JavaScript code, elements like NOFRAMES, NOSCRIPT, LABEL but the original layout is not affected.
- 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.
- Whether the representation language of the transformed page is the same as the original page or not. For example transforming an HTML page into a WML or cHTML page.
- If the transcoder is robust enough to handle correctly HTTPS, cookies, authentication, GETs and PUTs with all possible sets of parameters.
- If the transcoder is robust enough to handle badly written HTML, CSS or JavaScript code.
- The robustness of the annotations that allow the transcoder to provide customized transformations. Annotations 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, and produce some output. If the source code changes then it is likely that some annotation is not triggered anymore, or if it is, then it produces the wrong output.

There are also tools that support a model–based deeper transformation of the user interface to suit different devices (*e.g.* based on retargeting or on plastic user interfaces [19, 18, 7]) rather than simply translating the more external layer of a user interface. These tools are still confined to testing labs, and their analysis goes beyond the scope of this paper.

Mankoff and colleagues [16, 12] survey automatic transformation tools that can be either client–side or server–side (like transcoders). Authors discuss the support that transformation tools can offer to disabled users with a very limited human bandwidth (*e.g.* people having to use *sip–and–puff* interfaces), and index the features of these systems by disability. In [16] they discuss 7 requirements that transformation tools should satisfy (*e.g.* users can traverse the history list forward and backward, users are given alternatives for entering text and dealing with form elements) and present a browser prototype that implements some solutions (*e.g.* by automatically adding large buttons for paging up and down through sections of a web page).

LIFT Text Transcoder (LTT) is the text transcoder we used in our study. The goal of LTT [21] is to produce high quality text–only pages, that are 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 and PUTs but it does not support JavaScript), 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[1]. 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, mostly selectable by the end–user via visible options as are a few alternative color–themes.

MUFFIN [3, 2] is a Java–based, server–independent HTTP proxy, available through GNU GPL, 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 to 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` [14, 13] 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 automatically strip–off images, JavaScript or multimedia contents. It offers to the end–user a menu of different viewing options, one of which produces a text–only view of the page that 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

[1]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 (the target of the annotation), 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 transformation section says how to transform target elements matching the expression and belonging to documents on which the annotation applies. For additional details see [4].

moving navigation bars to the bottom of pages. We believe this transcoder supports HTTPS, GETs and PUTs. By being an Apache module, and thanks to the 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`).

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 (it treats data and layout tables in the same way), 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 is applied to. 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. We believe BETSIE does not support explicit filters.

`textualise` [9] is a transcoder whose goal is to make the text–only pages more accessible. Just 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. From the available demo one can see that it moves some navigation bars to the bottom of the page, it strips off all images, JavaScript and multimedia (and for some obscure reason it doubles all the content of the page).

The IBM transcoder described in [10] is a proxy server that is able to filter text, images, JavaScript and multimedia according to end–user preferences. The end–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, and appears to be somewhat compliant with respect to the MPEG-7 and CC/PP standards. In [17] authors present the *InfoPyramid* model that describes 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 to determine the level of details that is appropriate for the target device.

Access Gateway [5, 6] is a proxy server aimed at producing web pages that can be customized by the end–user (supposedly a person with low– or no–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.

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. The transformations are either built–in ones or they can be customized by the transcoder administrator (called filters or annotations).

Transcoded pages depend on three factors: (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 changes can be determined 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:

- transcoded pages are stripped of all the original CSS and JavaScript code, and simple CSS 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 (through annotations);
- scripts and event handlers are removed; when appropriate they are replaced with content of NOSCRIPT clauses;
- frames and framesets are replaced by corresponding contents; frames can be viewed simultaneously;
- layout tables are removed and their content is linearized, whereas data–tables are preserved;
- 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);
- 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

The purpose of the study is to determine whether text–only pages that are dynamically created through text transcoders are beneficial to disabled users. We framed this question into a comparative experiment aimed at measuring usability[2] under two experimental conditions: use of the original web site *vs.* use of the web site through an appropriately customized text transcoder.

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.

To better understand our problem, we ran two experiments. In the first one, 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 users preferences and performance. However it was quickly discovered that several factors were working against us, 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 the actions; others were moving across pages by using only 4 strokes, 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.

3. A large variability of user skills about accessibility features. For example no user took advantage of the *skip to page contents* link that would allow them to jump over long navigation bars.

4. Minor differences in the way in which links were shown 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).

On the basis of what we learned in this first user testing experiment, we conducted a larger test on a different web site. The test site belongs to an local Italian government agency[3] 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 [20], links with non–unique text labels).

3.1 Customization of the transcoder

Annotations for LTT were limited to process existing content of web pages and smooth out some of the accessibility barriers. The only new content added through annotations 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, like those specified by accessibility guidelines, pages were not customized to fit particular needs. Figure 1 shows a fragment of a transcoded page of the test site (seen through a browser running on a PC).

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).

We customized LTT in a way that is limited to process existing contents of web pages and smooth out some of the accessibility barriers mentioned above.

Figure 1: Screenshot of the page `www.regione.fvg.it/welcome.asp` produced by the transcoder. 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 with HTML headings H1, H2, H3).

[2]We adopted the ISO definition of accessibility: "*usability of a product, service, environment or facility by people with the widest range of capabilities*"[11].

[3]`www.regione.fvg.it`

3.2 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. See figures 2 and 3. As shown in figure 4, half of the participants use the PC for at least 30 hours/week, 25% use it for less than 12 hours/week; as for Internet usage, half of them use it for at least 10 hours/week, 25% for no more than 3 hours/week.

The sample of participants is therefore quite heterogeneous, and representative of a large population of disabled Internet users.

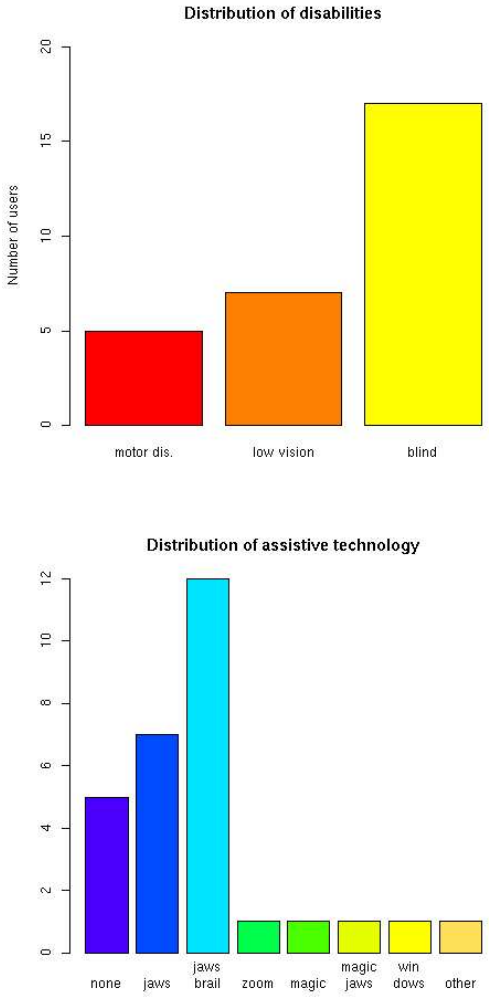

Figure 2: Distribution of participants according to type of disability and of assistive technology

Each participant was asked to carry out 5 information finding tasks, two with LTT, two without LTT and one with another transcoder [6]. The treatment (with/without LTT) and the order of the tasks were randomized to balance the learning effect. The first two tasks (being very similar) were

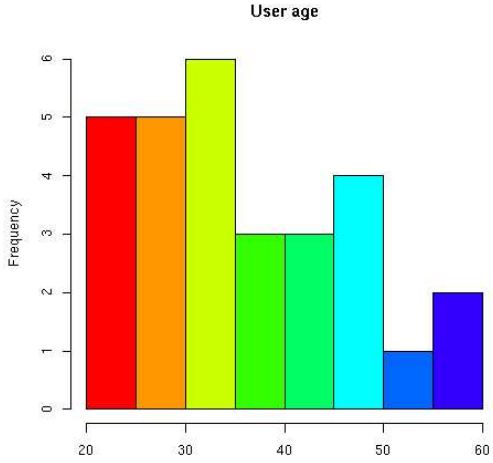

Figure 3: Distribution of participants according to age

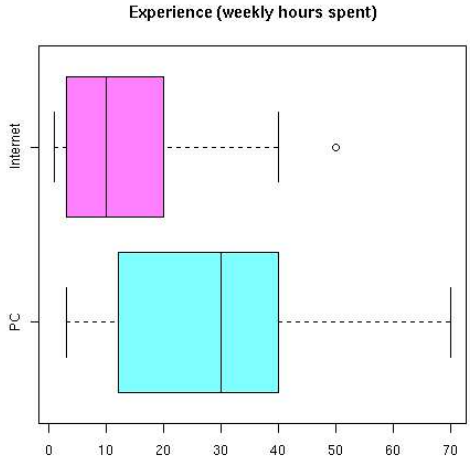

Figure 4: Distribution of participants weekly hours of use of PC and Internet. In the boxplots, whiskers represent the minimum and maximum values (including outliers); boxes represent the medians and quartiles.

simpler than the latter two (also similar to each other), that involved a form–filling activity, a typical accessibility barrier.

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

The 5^{th} task was needed in order to reduce the test effect by simulating a single blind test: participants did not know which transcoder we were studying, and we believe that this limited their bias when expressing their opinions. The outcomes of this task were not used to draw any conclusion.

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.

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 table 1, which presents also the statistically significant results.

4. RESULTS

4.1 Analysis of results

Visual inspection of the data suggests that none is normally distributed; therefore the null hypotheses (that LTT does not affect a certain parameter) had to be evaluated with non parametric tests, like the Wilcoxon rank test for paired samples. This is a relatively weak statistical test; therefore the figures we will soon present are conservative.

Table 1 shows the data produced by the experiment, along with significance figures where appropriate.

Analysis of correlation between variables shows that[4]:

- Q_2 and Q_1 show a marked correlation ($\rho = -0.75$ and -0.60, for simple and complex tasks respectively);
- Q_1 and Q_4: $\rho = 0.70, 0.73$;
- NP and NE: $\rho = 0.72, 0.69$;
- Q_1 and NE: $\rho = 0.70, 0.34$;
- CL and Q_4: $\rho = -0.56, -0.44$;
- CL and NE: $\rho = -0.48, -0.22$;
- CL and T: $\rho = -0.39, 0.19$.

By inspecting the differences under the two treatments, also the variables Q_1, Q_7, CL, GU, NP, T, NPS, NE and EPH agree and thus confirm the higher difficulty in the second pair of tasks.

These data confirm on the one hand the reliability of the questionnaire, and choice of tasks; on the other hand they show that the completion level is correlated to the satisfaction level, to the number of wrongly visited pages, and in part to the time required to complete the tasks.

[4]We 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.

Effectiveness

- LTT improved Q_1 and Q_2, 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*).
- LTT did not improve Q_3, for which no significant difference was found.
- LTT improved Q_4 for simple tasks and overall, but not for the complex tasks, for which no significant difference was found.
- 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.
- For more than 70% of the users, LTT would be the preferred choice for the next visit to the web site (Q_8).
- In terms of CL, more than half of the participants that used LTT reached the 100% success level (for complex tasks, and overall; no difference was found for simple tasks), whereas without LTT the median is 0.5. Also the means show a marked improvement of about 50%.
- Similarly for PR, the proportion of participants 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%.
- GU, the proportion of users that gave up, for complex tasks changes from 51% to 24% when using LTT; no difference was found overall nor for simple tasks.
- 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 found either for EPH, the number of errors per hour.
- 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.

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.
- 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.

	variable	factor	type	NO LTT mean (sd)	LTT mean (sd)	NO LTT median	LTT median	p–value
Q_1	*I easily found the required information*	E,S	simple	3.5 (1.68)	2.2 (1.45)	4	2	< 0.001
		E,S	complex	3.8 (1.68)	3.1 (1.45)	4	2	< 0.048
		E,S	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*	E,S	simple	2.6 (1.84)	3.6 (1.88)	1	5	< 0.033
		E,S	complex	2.2 (1.65)	3.4 (1.80)	1	5	< 0.003
		E,S	overall	2.40 (1.75)	3.5 (1.83)	1	5	< 0.002
Q_3	*I always knew where I was within the site*	E,S	simple	2.07 (1.41)	1.69 (1.19)	1	1	NS
		E,S	complex	2.13 (1.41)	2.03 (1.27)	1	2	NS
		E,S	overall	2.13 (1.40)	1.86 (1.23)	1	1	NS
Q_4	*I'm satisfied with the solution I found*	E,S	simple	3.1 (1.88)	2.0 (1.37)	3	1	< 0.005
		E,S	complex	3.4 (1.76)	2.8 (1.87)	4	2	NS
		E,S	overall	3.2 (1.81)	2.3 (1.68)	4	2	< 0.002
Q_5	*Assess the effort required*	P,S	overall	2.5 (1.32)	4 (1.03)	2	4	< 0.001
Q_6	*Evaluate the presentation quality*	S	overall	3.5 (1.4)	5.4 (1.18)	3	6	< 0.001
Q_7	*Rank the tasks by difficulty*	E	simple	2.38 (1.11)	3.24 (1.02)	3	4	< 0.008
		E	complex	1.97 (1.05)	2.38 (0.94)	2	2	NS
		E	overall	2.17 (0.55)	2.81 (0.54)	2	3	< 0.004
Q_8	*Proportion of users that would choose ... 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
		E	complex	0.45 (0.39)	0.56 (0.40)	0.4	0.4	NS
		E	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
		E	complex	0.41	0.48			NS
		E	overall	0.65	0.79			< 0.013
GU	Proportion of tasks when completion reason = *subject gave up*	E	simple	0.41	0.21			NS
		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
		P	complex	5.0 (3.02)	5.0 (2.31)	5	5	NS
		P	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
		P	complex	537 (360)	591 (417)	451	426	NS
		P	overall	501 (356)	446 (379)	446	349	NS
NPS	Number of visited pages when success=100%	P	simple	3.56 (1.33)	2.95 (0.76)	3	3	NS
		P	complex	8.14 (3.43)	5.63 (1.96)	7	5	< 0.041
		P	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
		E	complex	2.76 (2.41)	2.17 (2.28)	2	2	NS
		E	overall	2.67 (2.82)	1.57 (2.04)	2	1	< 0.014
EPH	Number of wrongly visited pages per hour	E	simple	14.6 (14.07)	18.79 (55.81)	12.7	0	NS
		E	complex	16.3 (13.46)	13.97 (13.72)	16.25	14.44	NS
NB	Number of clicks on *back*	E	simple	1.93 (2.04)	0.59 (0.83)	1	0	< 0.012
		E	complex	1.96 (3.34)	2.21 (4.17)	1	1	NS
		E	overall	1.95 (3.16)	1.40 (3.09)	1	0	< 0.045

Table 1: Dependent variables, their associated usability factor (Q_i means question, *E*,*P* and *S* stand for effectiveness, productivity and satisfaction, NS stands for *no significant difference*). Results are given in terms of means, standard deviations, medians and significance values. *Type* represents the type of tasks being used: simple, complex or both. Statistical significance was tested with the Wilcoxon test for paired samples; the maximum accepted significance level is 5%. 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

- For the mean time to complete successfully 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.

Satisfaction

- As commented above, LTT generally improves Q_1, Q_2, Q_4 and Q_8.

- LTT worsens Q_5.

- 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 background and foreground.

- Even though transcoded pages had the *skip–links* links, almost no user activated it; but most of them complained about the repetition of the same links over and over.

- Even though in many cases the linearization of the page content has helped users in locating the needed information, they often *forgot* to scroll it vertically, with the result that large portions of pages were not easily 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.

4.2 Discussion

The differences under the two treatments are not completely uniform (*i.e.* when no difference can be proven, this occurs mostly for the complex tasks), and this perhaps suggests that LTT, and the accessibility–based 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.

There is agreement between effectiveness and perceived effectiveness: on one side CL, PR, GU, NE, NB all agree (by considering the changes in the medians) and correlate well with Q_1, Q_2, Q_7 and Q_8.

Although there are weak differences to be reported for T (the time to complete a task), stronger differences arise for variables measuring effort (Q_5, NP and NPS), that agree.

There is also agreement between satisfaction and effectiveness: Q_4 and Q_8 agree with CL and PR.

Q_3 shows no significant difference, 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, when possible, did not add new content to the web pages, and therefore the original architecture was included also 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.

We weren't able to detect any usability difference due to the addition of access keys, of *skip–links* links and table of contents, that were very seldom being used.

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 3.2), since our sample of participants was not random. 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 well be the case that for visually unimpaired persons the effect of text transcoders could be radically different.

The differences should also apply to information web sites that require browsing, and for tasks similar to the ones we used in the experiment: fact finding with possibly one form to be filled.

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[5]. On the contrary, a text transcoder configured to fix the most common accessibility barriers tends to improve users effectiveness, productivity and satisfaction despite the removal of content and the automatically generated page layout.

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

[5]That share the same characteristics of the sample we studied, described in section 3.2.

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 that differ from those described in section 3.2, 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, in principle, we would expect to find similar results also with other text transcoders, provided that they support these kinds of transformations.

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.

Acknowledgments

Many thanks to Luciano Picone for his administrative help. We thank also Enrico Scoda and Francesca Leon for their help in developing annotations. We are also very grateful to all the participants whose time and effort were essential for this study. Many thanks also to Shadi Abu–Zahra for his focussed early comments. This work was partially funded through a grant by the Provincia di Udine.

6. 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. Stephanidis, editor, *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] 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`.

[6] S. S. Brown. About the web access gateway. `www.cl.cam.ac.uk/~ssb22/access.html`, 2004. Visited Nov. 2004.

[7] 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.

[8] CNIPA. Studio sulle linee guida recanti i requisiti tecnici e i diversi livelli per l'accessibilità e le metodologie tecniche per la verifica dell'accessibilità (legge 4 del 2004, art. 11 comma a e b). `www.pubbliaccesso.gov.it/biblioteca/documentazione/studi` Dec 2005. Visited Jan. 2005.

[9] codix.net. Web accessibility: textualise. `aquinas.venus.co.uk/solutions/products/textualise/index` Visited Nov. 2004.

[10] IBM Research. IBM transcoding technology. `www.research.ibm.com/networked_data_systems/transcoding` Visited Nov. 2004.

[11] 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`.

[12] 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/v01i03a`

[13] N. Kew. WebThing home page. `apache.webthing.com/mod_accessibility`. Visited Nov. 2004.

[14] N. Kew. mod_accessibility for Apache 2. `www.miswebdesign.com/resources/articles/mod_accessibili` 2003.

[15] S. MacKenzie. *Fitts' law as a performance model in human-computer interaction*. PhD thesis, University of Toronto, Ontario, Canada, 1991. `www.york.ca/mack`.

[16] 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.

[17] R. Mohan, J. Smith, and C.-S. Li. Adapting multimedia internet content for universal access. *IEEE Transactions on Multimedia*, pages 104–114, March 1999.

[18] 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 Academics Publishers, May 2002.

[19] 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.

[20] U.S. Dept. of Justice. Section 508 of the Rehabilitation Act. `www.access-board.gov/sec508/guide/1194.22.htm`, 2001.

[21] UsableNet Inc. LIFT Text Transcoder. `www.usablenet.com/products_services/text_transcoder/tex` Feb 2005. Demo version available at `demott.usablenet.com:8080/tt`.