

Accessing Google Docs via Screen Reader

Maria Claudia Buzzi¹, Marina Buzzi¹, Barbara Leporini²,
Giulio Mori¹, and Victor M.R. Penichet³

¹ CNR-IIT, via Moruzzi 1, 56124 Pisa, Italy

² CNR-ISTI, via Moruzzi 1, 56124, Pisa, Italy

³ Computer Systems Department, University of Castilla-La Mancha, Albacete, Spain
{Claudia.Buzzi, Marina.Buzzi, Giulio.Mori}@iit.cnr.it,
Barbara.Leporini@isti.cnr.it, victor.penichet@uclm.es

Abstract. Groupware systems allow remote collaboration via computer in a simple, economic and efficient way. However, to be universally valuable, groupware systems must be accessible and usable for all, including the differently-abled. In this paper we discuss the results of testing the accessibility and usability of Google Docs (<http://docs.google.com>) when using a screen reader and a voice synthesizer, and suggest some basic guidelines for designing effective, efficient and satisfactory User Interfaces (UIs).

Keywords: Accessibility, usability, groupware, screen reader, blind.

1 Introduction

In the early 1990s groupware systems transformed the way people worked together toward a common goal, allowing them to collaborate remotely via computer in a simple, economic and efficient way. In a few years these collaborative tools spread from organizations to the Internet. This opened up extraordinary opportunities for creating new world-wide collaboration tools, of which Wikipedia is one of the best known.

Collaborative tools range from classic email, blogs, chats, calendars and wikis to groupware systems such as eGroupware (<http://www.egroupware.org/>) or Google Docs (<http://docs.google.com/>). All these systems enable cooperation and increase productivity; however to be universally valuable, groupware systems must be accessible and usable for all, including the differently-abled. Accessibility is a pre-requisite for users to reach on-line application content, while usability offers simple, efficient and satisfying navigation and interaction. In order to evaluate whether a certain service/product is usable for those with special needs, it is first necessary to ensure that it is accessible.

Various accessibility and usability guidelines have been proposed in the literature to provide design criteria for developing Web contents. One of the more authoritative sources is the World Wide Web Consortium (W3C, Web Accessibility Initiative group), which defines accessibility guidelines for Web content, authoring tools and user agent design. The W3C Web Content Accessibility Guidelines (WCAG) are general principles for making Web content more accessible and usable for people with disabilities [19]. The latest version of the WCAG (2.0) is organized in four

categories, relying on the clear perception of content information (*content perceivable*), complete interaction with an interface in its functionalities (*interface elements operable*), comprehension of meaning (*content understandable*), and maximizing the interface's compatibility with new assistive technologies and devices (*content robustness*) [19]. Accessibility is closely related to usability; an interface should permit the user to achieve her/his target goal (*effectiveness*) in the best (*efficient*) and in a fully satisfactory way [5]. Important properties for website usability concern navigability as well as interaction. An interface should fulfill peoples' needs (*utility*), be easy to use in a gradual learning process for users who visit the site repeatedly (*learnability and memorability*) and limit user errors (*few easily recoverable errors*) [10].

In this paper we analyze the accessibility of Google Documents via screen reader and suggest some basic design guidelines. Section 2 briefly illustrates issues regarding interaction via screen reader and Section 3 presents some related works. Section 4 describes exploring Google Docs via screen reader, highlighting potential problems, and finally Section 5 introduces a discussion and future work.

2 Interacting via Screen Reader

Blind people perceive page content aurally and sequentially when accessing the Web by screen reader and voice synthesizer. Furthermore, blind users navigate mainly via keyboard. This kind of interaction with pages and user interface (UI) leads to several problems in perceiving content. Specifically, screen reader interaction causes:

- Content serialization. Content is announced sequentially, as it appears in the HTML code. This process is time-consuming and annoying when parts of the interface (such as the menu and navigation bar) are repeated on every page. As a consequence, blind users often quit a screen reading at the beginning, preferring to navigate by Tab key from link to link, or explore content row by row via arrow keys.
- Mixing content and structure. With Web content, the screen reader announces the most important interface elements such as links, images, and window objects as they appear in the code. For the blind user, these elements are important for figuring out the page structure, but require additional cognitive effort to interpret it. For instance, if a table's content is organized in columns, the screen reader (which reads by rows) announces the page content out of order; consequently, the information is confusing or misleading for the user.

This can lead to problems with user perception, such as lack of context, lack of interface overview (if the content is not organized in logical sections), difficulty understanding UI elements or working with form control elements (if not appropriately organized for interaction via keyboard), etc. This is discussed in greater detail in [8]. Thus, when designing for blind users it is essential to consider the three main interacting subsystems of the Human Processor Model: the perceptual, motor and cognitive systems [2]. The cognitive aspect of an interaction is extremely important, since many learning techniques are relevant for people with normal vision but not for the visually-impaired. Thus, alternative ways to deliver content should be provided. Furthermore, a blind person may develop a different mental model of both the interaction and the learning process, so it is crucial to provide a simple overview of the system as well as content.

3 Related Works

Web 2.0 applications brought a new dimension to the Web, moving from simple text and images created by single programmers, to multimedia and dynamic content increasingly built by users in collaborative environments. In the same way, the complexity of interface layout also increased. Since groupware environments vary greatly as to functions, interfaces, cooperation schemes and more, it is difficult to generalize very specific findings, whereas it is easier to compare homogenous classes of groupware applications. Regarding usability of on-line contents available in the World Wide Web, Takagi et al. suggest spending more time on the practical aspects of usability rather than focusing on the syntactic checking of Web pages, since some aspects are difficult to evaluate automatically, such as ease of understanding page structure and interface navigability [15].

Cooperative environments are particularly useful in the educational field, where knowledge is cooperatively assembled. Khan et al. carried out a usability study on ThinkFree, a collaborative writing system, in an educational environment, with four novice and four experienced users. Specifically, authors compared ThinkFree to Google Docs by means of a user test with Think Aloud protocol, a post-test questionnaire to collect user feedback and interviews to validate gathered results. Although the effectiveness of ThinkFree for the proposed tasks was shown, efficiency and availability of resources were more limited than in Google Docs.

To identify groupware accessibility issues and analyze proposed solutions, Schoerberlein et al. [13] revised recent literature highlighting the need for future research. Authors observed that most articles address the needs of a specific category of differently-abled persons. In particular, the needs of visually impaired persons are analyzed, due to the object difficulties these persons' experience when interacting with a complex layout via screen reader [7], [16], [17], [1].

Groupware should be usable for all, while the blind user is often required to have considerable computer skill. For simplifying access to a popular groupware system (i.e. Lotus Notes), Takagi et al. developed a self-talking client for allowing the blind to access groupware main functions efficiently and easily, masking the user from the complexity of the original visual interface [16]. Recently Kobayashi developed a client application (Voice Browser for Groupware systems, VoBG) for enabling visually impaired persons who are inexperienced with computer technology, to interact with a groupware system that is very popular in Japan (Garoon 2). The VoBG browser intercepts Web pages generated by the groupware server, parses their HTML code and simplifies on-fly their content and structure in a more understandable format for the target users [7].

Baker et al. adapted Nielsen's heuristic evaluation methodology to groupware; by means of a usability inspection conducted by expert and novice evaluators, they showed that this methodology can also be effectively applied by novice inspectors, at low cost [1]. Ramakrishnan et al. [12] investigate usability assessment in "information management systems", groupware environments characterized by mostly user asynchronous usage, integrating and adapting Nielsen's usability heuristics.

Awareness, one of the main properties of a groupware system, is also one of the accessibility principles: a user would be able to perceive when portions of UI reload and by means of the screen reader, to know the associated event occurring (e.g. a new

person joining the chat, a new message arriving on the board, a new user working on the document, and so on). To attempt to fill this gap, the WAI group is working on the Accessible Rich Internet Applications specification (WAI-ARIA) to make dynamic web content and applications (developed with Ajax, (X)HTML, JavaScript) more accessible to people with disabilities [18]. Using WAI-ARIA web designers can define roles to add semantic information to interface objects, mark regions of the page so users can move rapidly around the page via keyboard, define live regions, etc. [18].

Thiessen showed an example of using WAI-ARIA to design and implement a chat, highlighting some limitations of live regions [17]. However this problem is common with emerging standards, since browsers and assistive technologies need to conform to the new specifications, and this takes some time before reaching stable implementations.

4 Exploring Google Docs via Screen Reader

4.1 Methodology

In this first phase of our research we decided to examine interaction via screen reader with Google Docs in order to evaluate the real opportunities for blind people to write a document collaboratively. In this paper we consider accessibility issues, since accessing the tool and its main functions is a pre-requisite to considering other usability issues. With this in mind, we examined the Google Docs log-in, the Main (after the log-in access) and Document Editing (to create/modify an item of 'document' type) pages. The pages were analyzed by the authors, who performed a usability and accessibility inspection.

The pages were navigated using the screen reader JAWS for Windows Ver. 10.0 (<http://www.freedomscientific.com>), commonly used in Italy [8], and both the MS Internet Explorer (IE) version 8.0 and Mozilla Firefox version 3.0.5 browsers. One author has been totally blind since childhood and uses the JAWS screen reader every day; thus she is very familiar with JAWS functions and is able to use advanced commands. The different experiences of the authors when using JAWS allowed us to simulate both novice and proficient users (using advanced screen reader commands).

Testing Google Docs via screen reader was organized in three steps: (a) sighted user interaction with monitor power off (to simulate the experience of the blind or novice screen reader users) and power on (to confirm what was perceived in the first step) (b) blind user interaction (carried out by the blind author of this paper), (c) open discussion of personal experiences to verify and refine the collected data.

4.2 Results

Results showed that several main functions are practically inaccessible via keyboard, so interaction with Google Docs can be very frustrating for blind users. Main accessibility and usability issues encountered can be summarized as follows:

1. Orientation on the interface is difficult (not *efficient* or *satisfactory*): the main functions for an editing tool, such as creating or accessing a document, are announced after many elements (see Fig. 1, parts 1 and 2).

2. Accessing basic functions is very difficult (not *perceivable* and *operable*): some control elements cannot be accessed by the blind because they are not standard (X)HTML, so they are not correctly perceived by the screen reader; thus some tasks are impossible to complete (not *effective*). For example ‘Offline’ and ‘Show search options’ links (in the home page) are built programmatically and ‘div’ elements are used for visual rendering, but the focus via keyboard is not provided. Similarly, when accessing a Document, important menus (file, edit, view, insert, format, etc.) are impossible to access.

Furthermore, style properties (font type or size, etc.) are not accessible (not focusable via Tab key) since it is not possible to reach the formatting toolbar, while bold, italic or underlined functions can be used only through shortcuts, for the same reason. Unfortunately Google Documents shortcuts are different from those of MS Word; consequently, additional cognitive effort is required of the blind JAWS user (who is familiar with the Windows environment). Moreover, most editing functions do not have associated access keys.

3. Control elements are unclear (not *understandable*): buttons, links, etc. have labels which do not provide clear explanations about their usage and meaning.
4. Working with the documents is nearly impossible via keyboard (not *robust*): the IE browser does not allow selecting a document by checkbox (see Fig. 1, part 4), preventing the activation of related functions (e.g. delete, rename, mark as unviewed, revisions, etc.). The Firefox browser does not have this problem.
5. There are other differences in the behavior of the two browsers. Navigating via Tab key in the main page with Firefox, the ‘Create new’ pull-down menu does not receive the focus and is not announced; the tree view in part 3 of Fig. 1 is announced saying it must be explored via arrow keys, but arrows do not work. With IE the ‘Create new’ element is announced without specifying the type (pull-down menu) and the tree view is not announced via Tab key, but with arrows the user may access its single elements. Also toolbar visual buttons (‘Delete’ and ‘Rename’) and visual pull-down menu in part 4 of Fig. 1 (‘Share’, ‘Folders’, ‘More actions’) and visual pull-down menu in part 4 of Fig. 1 (‘Share’, ‘Folders’, ‘More actions’, ‘Last Modified’) are not accessible using IE, while they are accessible with Firefox, with unpredictable blocking behavior after exit (using ESC key) from ‘Folders’ pull-down menu.

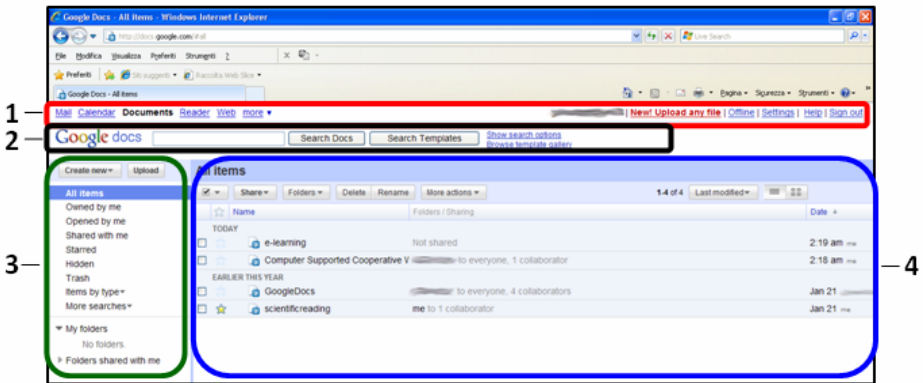


Fig. 1. Google Docs Home page, divided into functional areas

6. Tables used for layout purposes or for arranging the list of available documents have no summary attribute. In the main page of Google Docs IE announce a table without a summary, while Firefox does not announce it at all. A brief, functional and descriptive content could be assigned to the summary attributes in order to facilitate navigation through special commands like the letter “t” to quickly reach the next table, or to make the table content more understandable when reading sequentially via arrow keys. Furthermore, the ‘EARLIER THIS MONTH’ and ‘OLDER’ labels are not announced in both the browsers (when documents are ordered by date).
7. Dialogue windows are not accessible at all (Fig. 2).

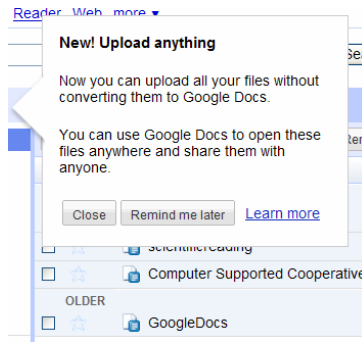


Fig. 2. An inaccessible dialogue window showing an informative message

4.3 Discussion

Groupware systems can be a great opportunity for people with mobility difficulties, such as blind users, in order to collaborate at a distance. However, if not appropriately designed, electronic barriers can prevent access to or enjoyment of all benefits of available resources. As reported in the previous section, several accessibility issues negatively affect potential use by the blind.

To have a perception “equivalent” to that of a sighted person, a blind user should know: a) where it is located on the interface, b) its status and c) what (s)he can do from that point. These three aspects are key factors in truly improving the overall efficiency of the interface. Feedback is not only important for user interaction with the system, but also to monitor the status of other users in the collaborative environment.

The W3C WAI-ARIA suite [18] resolves many of these issues. Indeed, WAI-ARIA adds semantics to interface objects (roles, states, properties) improving the interface control via screen reader. For instance, Web designers can define page regions, permitting the blind user to orientate him/herself in the UI (list of regions) and move rapidly from one section to another. WAI-ARIA allows marking sections specifying standard XHTML landmarks (main, navigation, search, banner, etc.) or defining regions, if they do not appropriately reflect the aim of the section. Especially, these customized regions can provide the user with a useful page overview. For example, the Google Docs home page could be organized into four main areas, representing different function groups: (1) applications and settings (2) search (3) file management

and (4) documents and properties (Fig. 1). Regarding awareness of what is happening in the UI, the definition of WAI-ARIA live regions enables the screen reader to intercept dynamic changes in the page (which does not reload the whole page, but only a portion, typical of Ajax applications) and informs the user.

Furthermore, to arrange a well-structured interface, a good approach would be to limit its complexity by dividing all commands or tools into functional groups, and providing the user with only those necessary for the current interaction, according to the user context based on the system focus.

Another facility provided by a well-designed interface is that the screen reader automatically enters Editing mode (e.g., when the focus is on a text edit widget) or Exploration mode (e.g., the focus is on a text-box or a link). In this way, for a blind user the learning process becomes more automatic and natural.

5 Conclusions and Future Work

In this paper we highlighted the main accessibility and usability issues that arise when interacting via screen reader with Google Docs. We analyzed the Google Docs home page and document editing to verify the accessibility and usability for blind people. Our analysis revealed issues that have a dramatic negative impact on the user interface and on user efficiency when carrying out basic tasks such as selecting and updating a document. We also formulated basic suggestions for improving the interaction and making user experience more satisfactory, allowing anyone to benefit from this collaborative tool, regardless of his/her ability.

In conclusion, we believe that design guidelines could have general applications, and that applying the WAI-ARIA suite would enhance usability via screen reader in any groupware environment. In the future we plan to evaluate how some WAI-ARIA properties applied to the Google Docs interface and its main functions can improve not only accessibility but also usability when interacting via screen reader.

References

1. Baker, K., Greenberg, S.: Empirical Development of a Heuristic Evaluation Methodology for Shared Workspace Groupware (2002)
2. Card, S.K., Moran, T.P., Newell, A.: *The Psychology of Human-computer Interaction*, pp. 29–97. Lawrence Erlbaum Associates, London (1983)
3. Davoli, P., Monari, M., Severinson Eklundh, K.: Peer activities on Web-learning platforms—Impact on collaborative writing and usability issues. In: *Education and Information Technologies*, September 2009, vol. 14(3), pp. 229–254. Springer, Netherlands (2008), doi:10.1007/s10639-008-9080-x
4. Huang, E.M., Mynatt, E.D., Russell, D.M., Sue, A.E.: Secrets to success and fatal flaws: The design of large-display groupware. *IEEE Computer Graphics* 26(1), 37–45 (2006)
5. ISO 9241-11. Ergonomic requirements for office work with visual display terminals (VDTs) - Part 11: Guidance on usability (1998)
6. Khan, M.A., Israr, N., Hassan, S.: Usability Evaluation of Web Office Applications in Collaborative Writing. In: *2010 International Conference on Intelligent Systems, Modelling and Simulation*, Liverpool, United Kingdom, January 27 - 29, pp. 147–151 (2010)

7. Kobayashi, M.: Voice Browser for Groupware Systems: VoBG - A Simple Groupware Client for Visually Impaired Students. In: Miesenberger, K., Klaus, J., Zagler, W.L., Karshmer, A.I. (eds.) ICCHP 2008. LNCS, vol. 5105, pp. 777–780. Springer, Heidelberg (2008)
8. Leporini, B., Andronico, P., Buzzi, M., Castillo, C.: Evaluating a modified Google user interface via screen reader. *Universal Access in the Information Society* (7/1-2) (2008)
9. Leporini, B., Paternò, F.: Applying web usability criteria for vision-impaired users: does it really improve task performance? *International Journal of Human-Computer Interaction (IJHCI)* 24(1), 17–47 (2008)
10. Nielsen, J.: *Usability engineering*. Morgan Kaufmann, San Diego (1993)
11. Petrie, H., Hamilton, F., King, N.: Tension, what tension? Website accessibility and visual design. In: 2004 International Cross-Disciplinary Workshop on Web Accessibility (W4A), pp. 13–18 (2004)
12. Ramakrishnan, R., Goldschmidt, B., Leibl, R., Holschuh, J.: Heuristics for usability assessment for groupware tools in an information management setting (2004)
13. Schoeberlein, J.G., Yuanqiong, W.: Groupware Accessibility for Persons with Disabilities. In: Stephanidis, C. (ed.) UAHCI 2009. LNCS, vol. 5616, pp. 404–413. Springer, Heidelberg (2009)
14. Schoeberlein, J.G., Yuanqiong, W.: Evaluating Groupware Accessibility. In: Stephanidis, C. (ed.) UAHCI 2009. LNCS, vol. 5616, pp. 414–423. Springer, Heidelberg (2009)
15. Takagi, H., Asakawa, C., Fukuda, K., Maeda, J.: Accessibility designer: visualizing usability for the blind. In: 6th International ACM SIGACCESS Conference on Computers and Accessibility, pp. 177–184 (2004)
16. Takagi, H., Asakawa, C., Itoh, T.: Non-Visual Groupware Client: Notes Reader. In: Proceedings of Center on Disability Technology and Persons with Disabilities Conference. California State University (2000)
17. Thiessen, P., Chen, C.: Ajax Live Regions: Chat as a Case Example. In: Proceedings of the 2007 International World Wide Web Conference, W4A (2007)
18. W3C. WAI-ARIA Best Practices. W3C Working Draft 4 (February 2008), <http://www.w3.org/TR/wai-aria-practices/>
19. W3C. Web Content Accessibility Guidelines 2.0, <http://www.w3.org/TR/WCAG20/> (December 5, 2008)