The HearSay Non-Visual Web Browser

Yevgen Borodin Jalal Mahmud I.V. Ramakrishnan Amanda Stent

Department of Computer Science
Stony Brook University
Stony Brook, NY 11794, USA
{borodin, jmahmud, ram, stent}@cs.sunysb.edu

ABSTRACT

This paper describes HearSay, an innovative non-visual Web browser, featuring context-directed browsing and an extensible VoiceXML dialog interface. The browser provides most of the standard browsing functionalities, including flexible navigation and form-filling. The paper also outlines future work aiming to make the Web more accessible for individuals with visual disabilities.

Categories and Subject Descriptors

H.5.2 [**Information Interfaces and Presentation**]: User Interfaces; H.5.4 [**Information Interfaces and Presentation**]: Hypertext/Hypermedia—*architectures, navigation*

General Terms

Design, Human Factors

Keywords

HearSay, Screen-Reader, Non-Visual, Browser, VoiceXML, Web Accessibility, Context-Directed Browsing, Navigation

1. INTRODUCTION

The Web is gradually becoming a primary source of information and is now taking a significant share of the retail market. Web sites are designed for graphical interaction, making Web browsing easy for sighted users who can quickly identify relevant information in Web pages. On the other hand, individuals with visual impairments have to use screen-readers to browse the Web. Current screen-readers, however, have plain interfaces sequentially verbalizing Web content. Our goal is to develop new technologies to meet the needs of the blind people and give them an easier access to information.

In this paper we present HearSay, a non-visual Web browser developed at Stony Brook University in collaboration with the Helen Keller Services for the Blind in Hempstead, NY. HearSay features a flexible dialog interface and innovative context-directed browsing [5] not used by any existing screen-readers. We are striving to implement the best features of existing screen-readers, but go beyond them in usability. HearSay brings together the fields of content analysis, natural language processing, and machine learning to help visually impaired and blind users browse the Web more efficiently and quickly identify relevant information in Web pages.

HearSay is a free open-source multi-platform browser written entirely in Java. It uses the Mozilla Web browser [6] to access the Internet and handle the standard browsing functionalities. HearSay uses the freely available FreeTTS [3] text-to-speech engine and Sphinx [8] speech recognition engine. It can be easily extended with new engines, and now also supports high-quality affordable commercial voices developed by Cepstral [2].

HearSay targets both blind and visually impaired users. Users interact with HearSay through a large input window and an extensible dialog system written in VoiceXML[1]. Non-visual navigation is treated as a dialog between a human and a Web browser. The dialog system is managed by VXMLSurfer [1] - a flexible VoiceXML interpreter developed in our lab. VXMLSurfer allows users to control HearSay with voice commands, text commands, or keyboard shortcuts. It can also handle multiple layers of dialog that give different views of the same Web page. Currently, HearSay uses only a basic screen-reading view.

Our initial ideas on HearSay appear in [7]. More details on the current architecture and algorithms are provided in [5]. The following section gives a detailed overview of HearSay features.

2. HEARSAY FEATURES

One of the highlights of HearSay is context-directed browsing, which allows users to avoid listening to Web pages from the beginning, and so to skip banners, ads, menus, etc. Instead, when users navigate from one page to another, HearSay starts reading the page from the most relevant section, thus, reducing information overload and potentially saving browsing time. It is a unique and innovative feature not used by any non-visual browsers. If HearSay misses the relevant section, it is very easy to go to the beginning of the page. Performance evaluation of contextual browsing is presented in [5].

[1] *VoiceXML* (VXML) is the W3C's standard XML format for specifying interactive voice dialogues between a human and a computer (www.w3.org/TR/voicexml20).

Users interact with HearSay through a large input window, which could be useful for individuals with low vision. It is possible to control the size and color scheme of the window to improve its visibility. The input window is used for entering text commands and form filling. HearSay can also echo user input.

HearSay uses our geometrical clustering algorithm [5] to partition Web pages into meaningful segments often containing semantically related information. The segments often correspond to menus, ads, tables, articles, etc., making it easy to navigate on most Web pages. HearSay allows users to move between these segments, as well as between the individual items and sentences on the page.

Hearsay has two modes of navigation: continuous-reading and pausing. In the continuous-reading mode, HearSay does not stop reading while the users navigate on the Web page. However, in the pausing mode, HearSay stops after reading each item or a sentence and waits for further instructions. The pausing mode requires more interaction on the part of users, but it is not as imposing as continuous-reading mode and, therefore, may be more suitable for novice users.

Besides moving between individual sentences and sections of the page, it is also possible to tab between form elements and links, as one would do when using a regular screen-reader. For each Web page, HearSay also provides menus containing all visited and unvisited links in the page. At any point, users can enable text copying, which makes HearSay append everything it reads into the clipboard.

To distinguish between different page contents (e.g.: links, buttons, images, etc.) HearSay uses either a verbal mode or an earcon mode. Earcons are sounds that are played every time a certain HTML element is encountered. Although earcons may take some time to get used to, the advantage of using earcons is in that they help distinguish between the actual page content and HearSay feedback. For example, if the word "link" occurs in a page, it can be confused with the system feedback saying "link" before each of the links on the page.

Hearsay also facilitates form filling. When a textbox or a text area are encountered, its value is read and is also displayed in the input window, where it can be edited. Radio buttons and combo-boxes are treated as lists providing an easy standardized way for filling HTML forms.

HearSay allows users to search Web pages for exact or approximate keyword matches. Approximate matching utilizes a part of the context-browsing algorithm [5] to locate the section of the page that is most relevant with respect to the search keywords.

HearSay is a multiuser Web browser keeping separate user-profiles that store different settings, history, and favorites for different users. Favorites, history, list of links, and other menus, are internally handled as subdialogs (similar to subprocedures) diverting the dialog flow from the main browsing dialog and then returning to the main dialog if needed.

HearSay supports any JSAPI [4] compliant text-to-speech engines and has controls to change between engines, voices, and voice properties (volume, rate, and pitch). The system can be paused at any time, and it also pauses if it goes off focus not to interfere with other accessibility software.

HearSay supports contextual help, which provides different help messages depending on the current state of the application. Because remembering all shortcuts may be cumbersome, HearSay supports both verbal and text commands. Most shortcuts have one or more corresponding commands, which are easier to remember or guess, e.g.: "help", "exit", "quit", etc. There are also other commands that are used less frequently, such as current "time", page "title", "link", etc. All of these are also accessible through the command menu, listing all shortcuts, commands, and their meaning.

The HearSay interface is flexible and easily extensible because all commands and shortcuts are treated as events, which have event-handlers in VoiceXML. Experienced users will be able to reassign or create new shortcuts, modify corresponding actions, change system feedback, create new menus, etc. We are continuously adding new features and are dedicated to improving the usability of the system. Users can leave feedback and send anonymous bug reports directly from HearSay by starting a feedback subdialog.

3. CONCLUSION AND FUTURE WORK

HearSay is an ongoing research project with plenty of room for usability improvements. While we are striving to introduce innovative ideas to facilitate non-visual browsing, we also understand the importance of being in touch with the end-users. We are consulting with the instructors of Helen Keller Services for the Blind and conducting evaluations to make sure that we meet the needs of the end-users.

Among our current projects there are handling JavaScript menus, HTML form analysis, table analysis, pattern-based partitioning, domain specific dialogs, transaction support, personalization, spell-checking, multilingual interface, support of Braille keyboards, etc.

4. ACKNOWLEDGEMENTS

We would like to thank the NSF (Award IIS-0534419) and CEWIT for supporting this research, and Cepstral.com [2] for donating realistic synthetic voices. We are grateful to the Helen Keller Services for the Blind for providing their invaluable feedback. And, finally, we would like to thank our developers for their dedication in developing HearSay.

5. ADDITIONAL AUTHORS

Psychology: Emmanuel Garcia emgarcia@ic.sunysb.edu
Computer Science: {id}@cs.sunysb.edu
Hui Guo {hguo}, Abhijit Aparadh {aaparadh}, Elton J. Pereira {epereira}, Rakesh Jawale {rjawale}, Padmini Patnaik {patnaik}, Asad Ahmed {asada}, Shreta Shah {shretas}, Rajat Gogri {rgogri}, Dhiraj Chawla {dchawla}, George John {gjohn}, Pranav Moolwaney {pmoolwan}.

6. REFERENCES

[1] Y. Borodin. A flexible VXML interpreter for non-visual web access. In *ACM ASSETS*, 2006.

[2] http://cepstral.com.

[3] http://freetts.sourceforge.net.

[4] http://java.sun.com/products/java-media/speech.

[5] J. Mahmud, Y. Borodin, and I. V. Ramakrishnan. Csurf: A context-driven non-visual web browser. In *International WWW Conference*, 2007.

[6] http://www.mozilla.com/firefox/.

[7] I. Ramakrishnan, A. Stent, and G. Yang. Hearsay: Enabling audio browsing on hypertext content. In *Intl. World Wide Web Conf. (WWW)*, 2004.

[8] http://cmusphinx.sourceforge.net.