# Towards One World Web with HearSay3

Yevgen Borodin
Stony Brook University
Comp. Sci. Department
Stony Brook, NY 11790
borodin@cs.sunysb.edu

Jeffrey P. Bigham
University of Washington
Dpt. of Comp. Sci. & Eng.
Seattle, WA 98195
jbigham@cs.washington.edu

Amanda Stent
Stony Brook University
Comp. Sci. Department
Stony Brook, NY 11790
stent@cs.sunysb.edu

I.V. Ramakrishnan
Stony Brook University
Comp. Sci. Department
Stony Brook, NY 11790
ram@cs.sunysb.edu

## ABSTRACT

In this paper, we present the key functionalities of *HearSay 3, a non-visual web browser* designed with the goal of improving web accessibility across the world. The featured functionalities include transparent support for multiple languages, collaborative labeling that harnesses the power of the social web, and a usable interface for handling dynamic web content.

## 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, Blind Users, Web Browser, Collaborative Accessibility, Multilingual Browser, Dynamic Content, Screen Reader.

## 1. INTRODUCTION

The Web has been designed primarily for sighted people since its very beginning. To assist users in browsing content designed to be viewed visually, we previously introduced the initial version of the HearSay web browser [7], which segmented pages into semantically meaningful parts. In version 2 of the browser, we used link context to identify and start reading from the page segment relevant with respect to the context [3,5]. The latest version of HearSay (version 3) addresses new problems brought to the forefront by Web 2.0.

For the past few years, the Web has evolved from a collection of static pages to a platform for interactive web applications. The popular screen readers have lagged behind the new technologies, resulting in widening of the accessibility gap between sighted and blind web users. The advent of Web 2.0 has increased problems as new technical challenges are introduced, coupled with ordinary web users contributing content. Technical standards are being developed that will allow web developers to make their web pages more accessible; however, these will only work if the developers choose to implement them. Currently, web access remains difficult, frustrating, and sometimes impossible for more than 11 million blind web users in the United States alone.

The world-wide impact of inaccessible web is even greater because accessibility problems are compounded with the lack of screen readers for many languages. For more than 45 million blind people across the world, the Internet has the potential to open access to information. A global non-visual browser must support multiple languages, just like its visual counterparts. Most of the current screen-readers cannot detect and automatically switch between languages. This is especially important for people who browse websites with mixed languages. The HearSay browser now offers support of several new languages and can detect and automatically switch between them. Not only will this be useful for multi-lingual users, but it may also help users wishing to learn to speak a foreign language.

As web pages are becoming more cluttered, using a serial audio interface to locate relevant information, links, and buttons in web pages has become more challenging. Navigating web pages would be easier if blind users had quick access to page elements that correspond to meaningful concepts, such as “item description”, “search”, “check out”, “navigation menu,” etc. Elements that lack accessible alternatives, such as images lacking alternative text, are also troubling; many sites that use image buttons do not provide the appropriate alternative text for them, making these sites inaccessible. Some screen-readers, such as JAWS [4] and aiBrowser [6], provide a facility for labeling content, allowing their users to personalize web pages. Thus far, these labels are confined to personal databases. Accessmonkey enables users to share scripts that improve accessibility, but these scripts can only be created by a programmer [2]. HearSay provides a flexible and accessible interface for collaborative labeling, in which users can share their labels with others.

The technology driving Web 2.0 has made possible real-time online collaborative services, such as Google documents, which requires handling of dynamic content delivered by AJAX. Although the W3C standard for Accessible Rich Internet Applications (ARIA - http://www.w3.org/TR/wai-aria) hopes to address this via content markup, most dynamic content available today does not conform to this standard. Observations of browsing patterns have shown that blind people tend to avoid dynamic web pages [1]. HearSay can now handle dynamic updates to web pages, providing a consistent view of the updates.

In the following section, we will overview the key features of the HearSay 3 non-visual web browser prototype.

*W4A 2008* - Challenge, April 21–22, 2008, Beijing, China.
Collocated with the 17$^{h}$ International World Wide Web Conference.

# 2. HEARSAY 3 KEY FEATURES

## 2.1 Multilingual Support

HearSay detects the languages used in a web page and chooses both an appropriate text-to-speech engine and a voice to read the page. Many websites specify the language with the XHTML *lang* attribute. When this attribute is included, HearSay 3 uses it to switch between the supported languages. In pages that mix multiple languages, such as wikipedia.org, the system seamlessly switches between languages as it reads the content. In future versions, we will use statistical algorithms for language detection and support more languages and text-to-speech engines.

## 2.2 Collaborative Labeling

HearSay enables users to assign labels to any web page elements using keyboard shortcuts or voice commands. HearSay gives its users the choice of either choosing a pre-existing label or defining a new one. The labels are stored in both local and remote repositories, in this way, maximizing the benefits of collaboration and personalization at the same time. The labels are "anchored" to the page elements by using a combination of XPATH (http://www.w3.org/TR/xpath) and element-specific attributes (such as the *src* attribute for images). When either the same user or another user visits a labeled page, the labels are retrieved from the repositories and applied on the page where appropriate. Labels found in a user's personal repository always take precedence over the shared labels.

We are currently researching collaborative filtering approaches that will mediate conflicting or competing labels. In addition, user labels can serve as training examples for machine learning algorithms, enabling them to leverage a few examples to automatically detect unlabeled concepts, and, thus, reduce the burden on users to provide labels manually.

Some web sites change frequently, resulting in periodic loss of labels. In this case, users will have to reassign the labels. However, as more people participate in collaborative labeling, each individual person will need to provide few labels. We are also building an accompanying visualization tool that will provide a visual look at the labels added by users and statistics concerning label usage. With this tool, web developers will be able to judge which labels are most important and optionally choose to integrate them into the design of their pages using appropriate web ontologies designed for accessibility.

## 2.3 Support of Dynamic Content

HearSay 3 detects dynamic updates (DHTML, AJAX, etc.) to the web pages and provides a usable interface to assist users in reviewing updated content. This enables HearSay to provide users with a consistent context when a page is dynamically updated. HearSay plays a background earcon to inform the user that content has changed with minimal interruption to the user's current task. Using a mixed-initiative design, the system does automatically jump to updated content, but, rather, lets the users review updated content at their convenience, and then optionally return to the original location they were viewing. Future versions of Hearsay will be fully compatible with ARIA, however, HearSay can already handle most types of dynamic updates without relying on web developers.

# 3. CONCLUSION AND FUTURE WORK

HearSay 3 is the next-generation self-voicing web browser designed to both overcome many of the current limitations of non-visual use of Web 2.0 applications and leverage its strengths to support collaborative accessibility through labeling.

Future versions of HearSay will continue to improve on both the features outlined here and on new features that we develop as part of our ongoing work with blind web users. We plan to release the beta version of HearSay in May 2008. This version will incorporate the features described in this paper and will be capable of working seamlessly with the JAWS screen reader.

We plan to evaluate the collaborative labeling feature with blind users to measure their ability to label and access labeled content. We will also evaluate the browsing efficiency when using the labels provided by other users. Their qualitative reactions to the labeling interface are as important as the technical aspects of the system, because the success of this collaborative approach, in the end, depends on the willingness of users to contribute labels. Previous approaches have relied on web developers who had minimal personal stake in achieving accessibility. In contrast, we believe that we have properly aligned the incentive to label content in order to improve one's own browsing efficiency with the ability to share those labels with others.

# 4. ACKNOWLEDGEMENTS

We would like to thank the NSF (Award IIS-0534419) and CEWIT (http://www.cewit.org/) 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

{ID}@cs.sunysb.edu in no particular order: Amit Saurav {asaurav}, Praveen Yarlagadda {pyarlaga}, Rohit Raman {rraman}, Mukta Joshi {mukta}, Sankar Aditya Tanguturi {stangutu}, Rohit Galwankar {rohitsg}, Thandava Batta {tbatta}, Jaini Shah {jshah}, Pratik Shah {pratiks}, Soumen Bandyopadhyay {sbandyop}, Janit Pal Singh Ahluwalia {janit}, Reema Sardana {reema}, Prakash Ahuja {pahuja}, Sneha Ramchandran {sneha}, Richi Gujral {rgujral}, Sushma Uppala {suppala}, Chunxiao Hou {cxhou}.

# 6. REFERENCES

[1] Bigham *et al.* WebinSitu: A Comparative Analysis of Blind and Sighted Browsing Behavior. ASSETS 2007.

[2] Bigham *et al.* Accessmonkey: A Collaborative Framework for Web Users and Developers. W4A 2007.

[3] Borodin *et al.* "The HearSay Non-Visual Web Browser". W4A 2007.

[4] JAWS 9.0. Retrieved February 2008.

[5] Mahmud *et al.* CSurf: A Context-Driven Non-Visual Web-Browser. WWW 2007.

[6] Miyashita *et al.* aiBrowser for Multimedia: Introducing Multimedia Content Accessibility for Visually Impaired Users. ASSETS 2007.

[7] Ramakrishnan *et al.* HearSay: Enabling Audio Browsing on Hypertext Content. WWW 2004.