Improving Non-Visual Web Access Using Context

Jalal Mahmud Yevgen Borodin Dipanjan Das I.V. Ramakrishnan

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

ABSTRACT

To browse the Web, blind people have to use screen readers, which process pages sequentially, making browsing time-consuming. We present a prototype system, CSurf, which provides all features of a regular screen reader, but when a user follows a link, CSurf captures the context of the link and uses it to identify relevant information on the next page. CSurf rearranges the content of the next page, so, that the relevant information is read out first. A series experiments have been conducted to evaluate the performance of CSurf.

Categories and Subject Descriptors

H.5.2 [**Information Interfaces and Presentation**]: User Interfaces—*natural language, Voice I/O*; H.5.4 [**Information Interfaces and Presentation**]: Hypertext/Hypermedia—*architectures, navigation*; H.3.3 [**Information Systems**]: Search and Retrieval

General Terms

Algorithms, Design, Human Factors, Experimentation

Keywords

Web Navigation, Context, Assistive Device, Screen-Reader, Voice Browsing, User Interface, CSurf, Non-Visual, Information Rearrangement, Partitioning

1. INTRODUCTION

Web sites are designed for graphical modes of interaction. Sighted users can visually segment Web pages and quickly identify relevant information, ignoring banners, navigation bars, etc. On the contrary, individuals with visual disabilities have to use screen readers, which process pages sequentially making Web browsing time-consuming. Searching and shortcut keys (e.g.: skip) offer only minor improvements. Unfortunately, in many cases, users still have to listen or skip through a substantial part of page content before they can get to the information; searching works only for exact string matching, disorienting users in case of a wrong match.

In this paper, we address this problem using the notion of *context*. Our prototype system, CSurf, provides all features of a usual screen reader. However, when a user follows a link, CSurf captures the context of the link and uses it to identify relevant information on the next page. CSurf rearranges the content of the next page, so, that the relevant information is read out first. We describe a technique for context identification, based on the structural and visual organization of Web pages. We present an algorithm and a system that can potentially help blind users to quickly identify relevant information on following a link, thus, considerably reducing their browsing time. The work described in this paper has broad connections to research in non-visual Web access, Web page segmentation, information rearrangement in Web pages, and contextual analysis. We implement the basic functionalities of a screen reader, extending the approach and the scope of Non-visual Web Access.

Some well-known screen readers for *non-visual Web access* are JAWS [3] and IBM's Home Page Reader [1]. BrookesTalk [9] makes summaries of Web pages. *Web page segmentation* techniques are either domain specific [7] or depend on fixed sets of manually specified rules [8], none of which are suitable for volatile Web sites. In contrast, CSurf geometric partitioning method is fully automated and scalable over domains. *Information Rearrangement in Web Pages* typically relies on either rules [6] or logical structures [4]. Our system automatically captures the contextual information and re-arranges the content to facilitate non-visual Web access. *Contextual Analysis* for non-visual Web navigation is not a well-studied problem. The system in [2] uses the context of a link to preview the next Web page before following a link. In contrast, we aim to help visually disabled users quickly identify relevant information *after* following a link.

ASSETS'06, October 22–25, 2006, Portland, Oregon, USA.
ACM 1-59593-290-9/06/0010.

2. SYSTEM ARCHITECTURE

The architecture of CSurf, our context enabled browsing system, is shown in Figure 1. Users communicate with the system through the **Interface Manager**. The module uses VoiceXML dialogs to interact with the users and present Web page content. In its current configuration, the system allows only keyboard input via our own VoiceXML interpreter. The Interface Manager provides both basic and extended screen-reader navigation features, such as shortcuts.

Context Analyzer is called twice for each Web page access. When the user follows a link, the module collects the context from the current page. When a new Web page is retrieved, the module executes our algorithm to contextualize the page before it is presented to the user.

The **Browser Object** module downloads Web content every time the user requests a new page to be retrieved. The module is built on top of the Mozilla Web Browser coupled with extended JREX Java API wrapper.

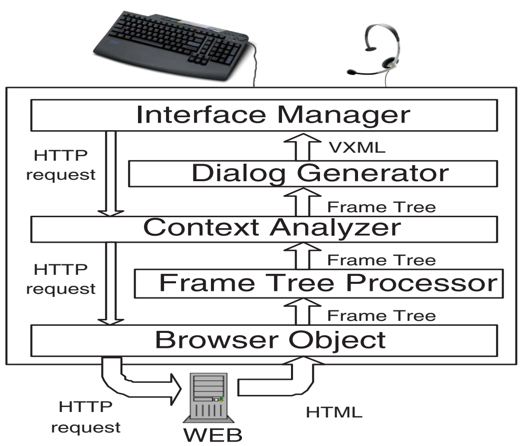

Figure 1: Architecture of CSurf

Frame Tree Processor extracts the *Frame Tree* representation of the Web page. A $Frame\ Tree$ is a tree-like data structure that contains Web page content and formatting, specifying how a Web page has to be rendered. The module cleans, reorganizes, and partitions the frame tree. Subsequently, Context Analyzer reorders the frame tree before passing it to the Dialog Generator.

The **Dialog Generator** module uses a collection of dialog templates to convert the frame tree into a Voice-XML dialog. The latter is then delivered to the Interface Manager. Some more architectural details appear in [5].

3. CONTEXT ANALYSIS

Context analysis is performed every time the user follows a link. The Context Analyzer module uses the frame tree of the source Web page to identify context around the link. It then utilizes this contextual information to identify relevant portions of text in the destination Web page.

In the **Context Identification** phase, the frame tree is searched for all nodes containing the URL selected by the user. The text in and around the link is extracted from the frame tree and stored in a multiset, after removing all function words. In the **Context Ranking** phase of the algorithm, the extracted words are ranked according to their proximity to the link. The destination Web page is then fetched. The **Context Matching** algorithm matches the words in the multiset to the text in all leaves of the new frame tree, which are then assigned respective weights. In the **Block Ranking** phase, we propagate the weights from the leaves of the frame tree up to a certain node/block, which we identify as a parent of semantically related nodes. The most relevant block of information will have the highest rank. At the **Block Rearrangement** phase, the frame tree is reorganized based on the weights of the blocks, so that the most relevant block of information is placed first.

4. EVALUATION

A series experiments have been conducted to evaluate the accuracy of CSurf as well as its quantitative performance against JAWS, the state-of-the-art screen reader. In our evaluation we used twenty four Web sites spanning four content domains: *news*, *books*, *consumer electronics*, and *office supplies*. We focused on the time taken to access the information with and without context-directed browsing. We allowed the use of JAWS shortcut keys to skip blocks of text, thus, accelerating browsing. Still, compared with JAWS, our system achieved browsing time reduction of 63.7% in news domain, 57.9% in books domain, 45.2% in electronics domain, and 54.4% in office supplies domain.

We calculated how well CSurf identified relevant information in each Web site. Our algorithm showed a reasonable accuracy of over 80% in all four content domains. It is notable that it performed the best in the news domain (88%). The structural and geometric organization is often much better in News Web sites than in other domains. Since our context analysis algorithm depends on the geometric organization of Web pages, it explains why CSuft performed better in the news domain. The results showed that our technique works best on well-structured, information-driven, dynamic Web sites such as news engines, encyclopedias, online stores, etc. The evaluation demonstrated that the use of context can save browsing time and substantially improve browsing experience of visually disabled individuals.

5. CONCLUSION AND FUTURE WORK

In this paper, we have described the design and implementation of CSurf, a context-directed non-visual Web browsing system. We presented experimental evidence of the efficiency of context-directed Web browsing using audio. The following are only a few potentially useful areas for further research. If search keywords are treated as context, our algorithm can be easily extended to allow smart searching within a Web page. NLP techniques can be employed to enhance context processing and searching. We are currently researching summarization techniques to further save the browsing time. We are also investigating the feasibility of applying machine learning algorithms and statistical models to context identification and ranking. Our context processing system can be further enhanced by making it browser- and screen-reader independent.

6. REFERENCES

[1] C. Asakawa and T. Itoh. User interface of a home page reader. In *ACM Intl. Conf. on Assistive Technologies (ASSETS)*, 1998.

[2] S. Harper, C. Goble, R. Stevens, and Y. Yesilada. Middleware to expand context and preview in hypertext. In *Assets '04: Proceedings of the 6th international ACM SIGACCESS conference on Computers and accessibility*, 2004.

[3] http://www.freedomscientific.com/.

[4] K. Nagao, Y. Shirai, and K. Squire. Semantic annotation and transcoding: Making web content more accessible. *IEEE Multimedia*, 8(2), 2001.

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

[6] T. Raman. Audio system for technical readings. *PhD Thesis, Cornell University*, 1994.

[7] H. Takagi, C. Asakawa, K. Fukuda, and J. Maeda. Site-wide annotation: Reconstructing existing pages to be accessible. In *ACM Intl. Conf. on Assistive Technologies (ASSETS)*, 2002.

[8] S. Yu, D. Cai, J.-R. Wen, and W.-Y. Ma. Improving pseudo-relevance feedback in web information retrieval using web page segnmentation. In *Intl. World Wide Web Conf. (WWW)*, 2003.

[9] M. Zajicek, C. Powell, and C. Reeves. Web search and orientation with brookestalk. In *Proceedings of Tech. and Persons with Disabilities Conf.*, 1999.