A Voice-driven Web Browser for Blind People

Boštjan Vesnicer, Janez Žibert, Simon Dobrišek,
Nikola Pavešić, France Mihelič

Laboratory of Artificial Perception, Systems and Cybernetics
Faculty of Electrical Engineering, University of Ljubljana
Tržaška 25, 1000 Ljubljana, Slovenia
bostjan.vesnicer@fe.uni-lj.si

Abstract

A small self-voicing Web browser designed for blind users is presented. The Web browser was built from the GTK Web browser Dillo, which is a free software project in terms of the GNU general public license. Additional functionality has been introduced to this original browser in form of different modules. The browser operates in two different modes, browsing mode and dialogue mode. In browsing mode user navigates through structure of Web pages using mouse and/or keyboard. When in dialogue mode, the dialogue module offers different actions and the user chooses between them using either keyboard or spoken-commands which are recognized by the speech-recognition module. The content of the page is presented to the user by screen-reader module which uses text-to-speech module for its output.

The browser is capable of displaying all common Web pages that do not contain frames, java or flash animations. However, the best performance is achieved when pages comply with the recommendations set by the WAI.

The browser has been developed in Linux operating system and later ported to Windows 9x/ME/NT/2000/XP platform. Currently it is being tested by members of the Slovenian blind people society. Any suggestions or wishes from them will be considered for inclusion in future versions of the browser.

1. Introduction

Modern information-technology facilities are often not suitable for blind and visually impaired people. Such problems in communication are well known to many disabled persons. If they are unable to use their hands, read or speak, they are forced to use technical aids to overcome their problems. For blind or visually impaired persons the Braille coding of texts is a common aid. This type of coding requires special editions of written corpora or special additional hardware components when used with computers. The solution is relatively costly and requires special skills from the user.

Over the past ten years a considerable advance has been made in the development of automatic text-to-speech and speech recognition systems. Such systems offer a more natural and user-friendly way of communication for the blind or visually impaired persons; the communication goal can be achieved faster and they offer access to large text corpora via modern technical equipment (over computer networks, scanners, etc.) and have a relatively low price [9].

However, these new technologies are very language dependent and general solutions for all languages cannot be applied directly [6]. If speech technologies are to be used with the Slovene language the language-dependent parts of the systems must be developed for this purpose using knowledge of Slovene-language phonology, syntax and semantics.

Spoken-language technologies have been one of our main research activities for more than twenty years. Our prime interest is to develop a core technology for the Slovene spoken language that could be customized for different kinds of applications. We found the development of a voice-driven Web browser for Slovene-speaking blind people important to our research for several technical and non-technical reasons, among them is the possibility to help the disabled people.

A voice-driven Web browser called Homer was developed for reading Slovenian texts obtained from the Internet site of the Association of Slovenian Blind and Visually Impaired Persons Societies.

2. The Homer Web browser

The Homer Web Browser presents the most recent phase in the evolution of the information retrieval system for blind and visually impaired people [3]. However, it was not built from scratch since new modules were just introduced into the source code of one of the publicly available Web browsers. When choosing among different browsers we compared their speed, stability and extensibility. We found that the GTK Web Browser Dillo [1], which is now considered to be yet another module of the whole Homer system, was perfect for our needs.

2.1. The Homer system structure

The whole system consists of five main modules (see Figure 1).

Input to the system is performed via a keyboard, with some specially selected keys or by using the speaker-independent spoken command recognition module that runs in parallel with the other modules. The voice control of the system additionally facilitates working with the system as there is no need to use mechanical interfaces. The dialogue module manages dialogues with users and performs access to the Web pages via the Web browser.

The most important part of the system is the first fully developed Slovenian text-to-speech system [4], which is essential for blind users. It enables automatic generation of Slovene speech from an arbitrary Slovenian text written in a non-tagged plain-text format.

The Homer browser was designed to run on a standard PC with a minimum of 64 MB of RAM with a built-in standard 16-bit sound card and a standard headset with a close-talking microphone. Initially it was developed for Linux platform

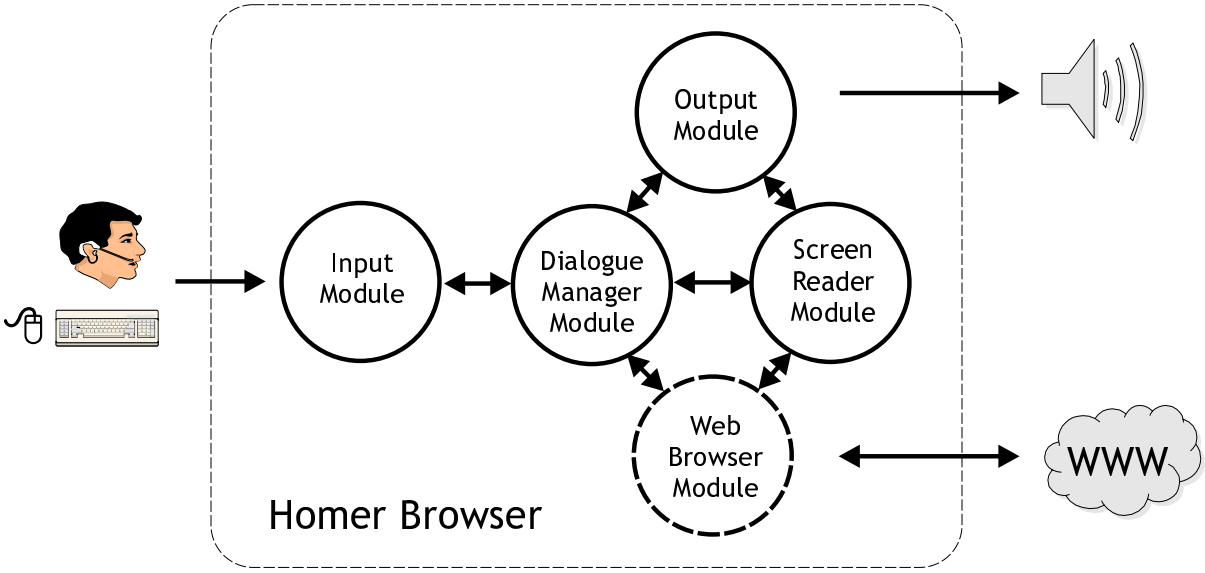

Figure 1: *The structure of the Homer system.*

and later ported to the Microsoft Windows 9x/ME/NT/2000/XP operating systems. For the best performance it uses multi-threading and other advantages of the 32-bit environment. It requires approximately 15 MB of disk space for the program code and for the text-to-speech and speech recognizer module inventory.

2.2. Screen reader module

Our first step was to add a screen-reader function to the existing Dillo source code. The built-in screen reader is now triggered by pointing the mouse and uses the text-to-speech module for its output. When a user stays for a moment at a certain position on the Web page the text beneath the pointer is sent to the output text-to-speech module. The output module works in a separated thread with a time-out function that prevents the user from overfilling the synthesis buffer with a fast pointer motion when browsing through the Web page.

An important feature of the screen reader is that it signals special events when user with a mouse pointer leaves or enters a particular area of the displayed Web page. These events are handled by the output module, which generates different sounds to inform the user about position of the mouse pointer.

The screen-reader function supports not only text parts of common Web pages but some basic graphic objects as well, such as non-animated images, lines, bullets, buttons and input text fields. When a user stays for a moment with a mouse pointer pointing at such graphic objects then the system sends a short description of the object to the output module. An example of such a description would be: *"Button labelled 'send', sized 60x20 pixels"*. As a result, blind user is able to browse not only common but also some more graphically intensive Web pages (see ~~Figure 2~~).

The screen reader works in several different modes. It can read individual words, sentences, lines and paragraphs of the displayed Web page. It can read page headings and the whole page as well. One can easily switch between different reading modes simply by pressing function keys on a standard PC keyboard.

2.3. Input module

The input interface manages the keyboard entry and/or spoken-command recognition. Each of the spoken control commands is associated with its accelerator key on a standard PC keyboard.

The speaker-independent spoken commands recognition module is based on tied-mixture continuous HMMs of fundamental phone transition-like units [2]. These models are used as the fundamental models in a silence/commands/silence HMM graph. A number of improvements to the acoustic modelling were introduced. The parameters of the fundamental models were estimated from the Gopolis spoken-language database [7], which contains several hours of speech from 50 speakers.

A preliminary off-line evaluation of the spoken-command recognition accuracy, using a clean speech database of ten test speakers, yielded an average recognition error rate lower than 2 %. However, the actual recognition rate is strongly dependent on the spoken-command grammar and the user's behaviour while interacting with the system. In practice, the online recognition error rate increases, but remains below 5 %.

2.4. Dialogue module

The dialogue module manages dialogues with users, accesses Web pages via the Web browser module and performs the system-control function. Its design is based on our experiences with the design of a similar dialogue module in another speech recognition system [5].

Since a structure of common Web pages can always be presented as a tree structure, the dialogue module enables transitions between all the tree nodes at any stage of the processing and navigating through a list of sub-nodes at each of the tree nodes as well. There are only three main actions that the dialogue module takes or offers to user. These actions are: opening of a Web page tree node, navigating through a list of sub-nodes, and closing a Web page tree node. Each of the tree nodes represents a link, a page itself, or a part of a page.

All the actions can either be just offered to the user or are taken immediately after the dialogue module successfully interprets a recognized spoken command or a keyboard entry, even

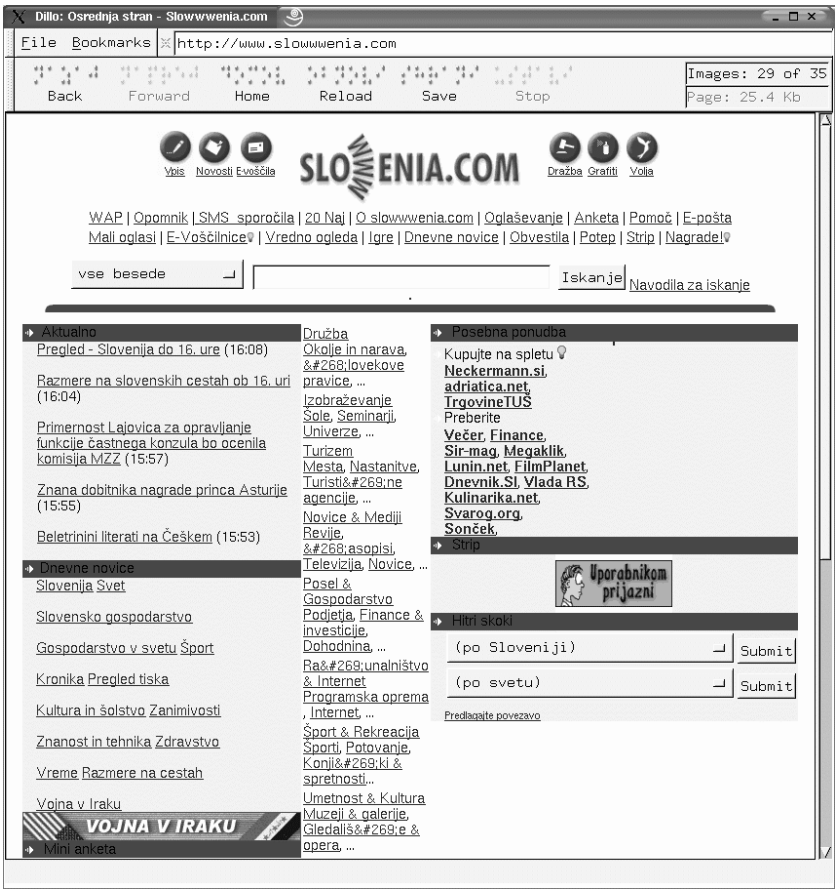

Figure 2: *An example screen snapshot of a more graphically intensive Web page (in Slovene).*

though the dialogue is still in the process of reading/describing a part of the Web page. As a result, these two different dialogue strategies were implemented. When an action is explicitly offered then a *yes/no* answer is expected from the user. This dialogue strategy is called *passive-user behaviour*. On the other hand, an *active-user behaviour* strategy means that the dialogue module expects the user to interrupt the dialogue process with spoken commands.

The first strategy is more suitable for beginners; the second is more suitable for expert users since it enables faster navigation. Both strategies have to be combined when the dialogue is in the process of reading a selected text. In this case the user is allowed to occasionally interrupt the reading process with commands.

The dominantly passive dialogue strategy requires only yes/no answers and a small number of easy-to-remember navigation commands. We found that this dialogue strategy provides a very comfortable interaction with the system, even though it has proved to be rather time consuming and even annoying to expert users.

The active dialogue strategy requires some additional navigation commands. In the best case we could use a dynamic spoken-command grammar instead of using just a static list of commands. The current version of the speech recognition input module did not allow us to use the dynamic spoken-command grammar, thus we carefully selected a small number of navigation commands. We found that the navigation speed is very comfortable when using eight basic commands. Their English translations are: “Open!”, “Close!”, “Skip!”, “Previous!”, “Repeat!”, “Restart!”, “Pause!”, and “Resume!”. With these commands a user can open and close Web page tree nodes and navigate through a list of sub-nodes at any position in the tree.

2.5. Output module

For the automatic conversion of the output text into its spoken form the first Slovenian text-to-speech system called S5 [4] based on diphone concatenation was applied. The non-tagged plain text is transformed into its spoken equivalent by several modules. A grapheme-to-allophone module produces strings of phonetic symbols based on information in the written text. A prosodic generator assigns pitch and duration values to individual phones. The final speech synthesis is based on diphone concatenation using TD-PSOLA [8].

Besides speech output, the output module (in connection with the screen-reader module) produces also special distinctive non-speech sounds, which inform the user about changes to the current position of the mouse pointer.

3. Conclusions and future work

The development of the Homer system is still in progress. We expect the system to evolve towards a specialized Web browser with a mouse-driven text-to-speech screen reader and a voice-driven dialogue manager that handles all common Web pages.

Improvements in the sense of more accurate and robust

speech recognition are planned for the future. Work on speech recognition that incorporates a larger dynamic spoken-command grammar is already under way. We are also expecting further suggestions from the blind and visually impaired community, especially with regards to the design of the strategy for communication with the system and, of course, remarks on the Slovene speech synthesis quality.

4. Acknowledgments

This work was partly funded by Ministry of Education, Science and Sport of Slovenia contr. nr. L2-2109 and HP Voice Web Initiative grant <http://webcenter.hp.com/grants/>.

5. References

[1] *"The Dillo Web Browser"*, <http://dillo.auriga.wearlab.de/>, 2003.

[2] Dobrišek, S., *Analysis and recognition of phones in speech signal*, Ph.D. thesis (in Slovene), University of Ljubljana, 2001.

[3] Dobrišek, S., Gros, J., Vesnicer, B., Pavešić, N. and Mihelič, F., "Evolution of the information retrieval system for blind and visually impaired people", *International Journal of Speech Technology*, 6(3): 301–320, 2003.

[4] Gros, J., Pavešić, N. and Mihelič, F., "Text-to-speech synthesis: A complete system for the Slovenian language", *Journal of Computing and Information Technology,* 5(1):11–19, 1997.

[5] Ipsić, I. and Pavešić, N., "An overview of the Slovenian spoken dialog system", *Journal of Computing and Information Technology,* 10(4):295–301, 2002.

[6] Jelinek, F., *Statistical methods for speech recognition*, Cambridge: The MIT Press, 1998.

[7] Mihelič, F., Gros, J., Dobrišek, S., Žibert, J. and Pavešić, N., "Spoken Language Resources at LUKS of the University of Ljubljana", *International Journal of Speech Technology*, 6(3): 221–232, 2003.

[8] Moulines, E. and Charpentier, F., "Pitch-synchronous waveform processing techniques for text-to-speech synthesis using diphones", *Speech Communication*, 9: 453–467, 1990.

[9] Zajicek, M., Powell, C. and Reeves, C., "Ergonomic factors for a speaking computer interface", In M. A. Hanson, E. J. Lovesey and S. A. Robertson (Eds.), *Contemporary Ergonomics – The proceedings of the 50th Ergonomics Society Conference,* Leicester University, London: Taylor and Francis, pp. 484–488, 1999.