USER INTERFACES FOR VISUALLY IMPAIRED PEOPLE

Robert Batůšek and Ivan Kopeček
Faculty of Informatics
Masaryk University
Botanická 68a, 602 00 Brno, Czech Republic
e-mail: xbatusek@fi.muni.cz, kopecek@fi.muni.cz,
http://www.fi.muni.cz/~xbatusek/, http://www.fi.muni.cz/~kopecek/

Abstract

Some aspects of the user interfaces and spoken language dialogue systems developed for visually impaired users are presented in the paper. First, we discuss specific requirements of visually impaired users to user interfaces of applications and dialogue systems. The paper further deals with some specific elements of the dialogue system design that can be used to enhance the effectiveness of the communication between blind user and dialogue systems.

Next, the basic communication module used in the dialogue systems we are presently developing is outlined. We also present a brief description of two applications developed by the Natural Language Processing Group at the Faculty of Informatics, Masaryk University Brno; the dialogue programming system DIALOG and the speech oriented hypertext system AUDIS. Both systems are developed primarily for visually impaired users, especially for blind programmers and students.

1. Introduction

Although present software systems are often very sophisticated and user-friendly, they are usually not very convenient for visually impaired people. The reason is in the graphical interface and absence of the features fulfilling special needs of the blind people. Speech synthesizer and "screen access" ("screen reader") software still represents basic facilities that are used by blind users to obtain information by means of computer.

The present development in human-computer interaction and spoken language dialogue systems (especially multi-modality and progress towards less error-prone speech recognition) brings new hopes and expectations and also new problems. The design of suitable dialogue strategies which seems to be one of the crucial points of the dialogue systems development is even more important for visually impaired people, supporting perspicuity of the visually impaired user. This aspect is emphasized by the fact that for visually impaired people computers are one of the most important sources of information.

2. Specific Demands on User Interfaces for Visually impaired

In some applications, there is almost no difference in using the user interface between sighted and visually impaired user. This is, for example, the case of dialogue systems that are accessible via telephone. However, many systems use graphics as an important output information and they mostly also do not assume that the user is visually impaired and therefore they ignore specific needs of such users. Let us first summarize basic specific needs of the visually impaired user:

- The system must enable comfortable control by means of combination of speech commands and keyboard (hot-key commands). Of course, graphics can be used only in a special way as an additional output for the users that are not totally blind. Other input/output devices can be used for a specific application provided that such a type of communication is effective.
- Speech commands should be supported by speech (system driven) command dictionary that allows to express a command in several ways, making the control of the system more intuitive. This is always useful, but especially for blind users where the possibility of intuitive control of the system is even more important.
- Easy customization and configuration is very important feature of the system, especially for blind users that often use the system for a long time. This is related to the control commands, mode and type of speech synthesis output, information data structure, and other properties and options of the system.
- It is very important to enable the user to obtain the information quickly and to allow to get an informational overview. This feature is supported by various output speech modes and output speech rates as well as by speech summaries, audio glances, earcons and environmental sounds.
- The orientation of the user should be supported by the information about the position that is always accessible in speech form as well as in the form of audio glances, earcons and environmental sounds (see next section).

The last two points are closely related to the problem of the design of dialogue strategy and we will discuss them in more detail below.

3. Specific Elements of the Dialogue Design of the Blind User Oriented Systems

As we have pointed out, one of the most important problems we meet when designing spoken language dialogue systems for blind users is how to convey a variety of pieces of information (some of them can take basically graphical form) quickly and how to supply sufficient information providing full orientation of the user. The main way how to manage this is, of course, to use sound. It can be done in the form of:

- Synthesized voice produced by the syllable based speech synthesizer. This type of sound output can be used for generating output messages and reading text data. The used speech synthesizer should apply basic prosodic features to enhance quality of the speech output and to distinguish various types of utterances. It should be also able to use various voice types that can be configured by the user. Various voice types can be used to distinguish various types of information.
- Sampled voice, which can be used for all feedback messages to the user. Various types of sampled voices are used to help the user to distinguish various types of messages.
- Sound generated by the sound synthesizer, MIDI, wave tables or special samples. This type of non-speech sound is used first for environmental sounds applied to provide feedback to user actions [2], [3], secondly it can be used for earcons (non-speech glances used to give the blind user an overview by listening, [9].

Non-speech sound and flexible use of various type of the speech type can essentially help to speed up the communication. On the other hand, it can also confuse the user, if he or she is not familiar with the corresponding meaning. Hence, we propose the strategy that consist in:

- The system has to find out whether it communicates with a beginner or experienced user. It can be done either by detecting it from user reactions or by user declaration;
- Depending on the established user experience measure the system chooses the corresponding strategy, i.e. communication level that combines explicit and implicit information.
- In any time a user can use the command EXPLAIN which explains the meaning of the implicit information and therefore learn it by using the system.
- The system checks the communication; if it detects that the user shows tendency to use EXPLAIN too frequently (or if the user does not use EXPLAIN at all), it switches the communication level.
- The user can switch off the above-described "regulation" to enable learning mode or to set the communication level.

4. The Structure of a Speech Interface for Blind Users

The communication strategy described above has to be supported by the module which forms the interface between the dialogue oriented application and hardware devices and speech synthesizer and recognizer tools. Structure of such module is outlined in Figure 1. The interface has modular structure and uses some previously developed tools: the TTS system DEMOSTHENES [6], command recognition module RCG and the prosody detection module [7]. The interface is designed to be reusable in various kinds of applications, especially in the DIALOG system (see next section) and other applications for visually impaired users.

An application communicates with the main module. It sends to the interface simple requests like *Say 'Hello, world'* or *Play 'asound.wav'*. The interface informs the application about executing its requests. The application can pause (and resume) speech at any time. The interface also sends messages to the application when it recognizes voice command.

The main module of the interface operates with the sound device and guarantees that it is opened for input and/or output whenever it is needed. An application requests to perform some operations by calling a simple function. The main module only processes the request and delivers it to the appropriate submodule. The control is then returned to the application. The submodule working in a separate thread executes the request and informs the main module about the current state of the execution. The main module can inform (depending on the configuration) the application using a message. The application can use a message handler to dispatch the message.

Command recognition module continuously "listens" to the input device (usually a microphone) and tries to recognize a command. When it detects a correct command stored in its database, it informs the main module. The main module generates the appropriate message and sends it to the application.

The main module also cooperates with the prosody detection module that distinguishes three types of intonation: rising, level and falling. A command spoken with the rising intonation is understood to be a demand for assistance. A word spoken with the level intonation is interpreted as a command that will be followed by a list of parameters (or one of these parameters when such a command already has preceded). A word pronounced with the falling intonation forms a command without parameters or the last command parameter.

Sound generation module processes the texts placed to the submodule queue by the main module. These tagged texts include requests for TTS system to synthesize a text, configuration of the synthesizer (changes of the current voice, rate, volume etc.), requests for playing audio files or for mixing audio and speech. Sound manager module can either load an arbitrary audio file or one of the predefined sampled sounds (or speech messages) stored in the internal module database. It could also synthesize sounds using soundcard capabilities.

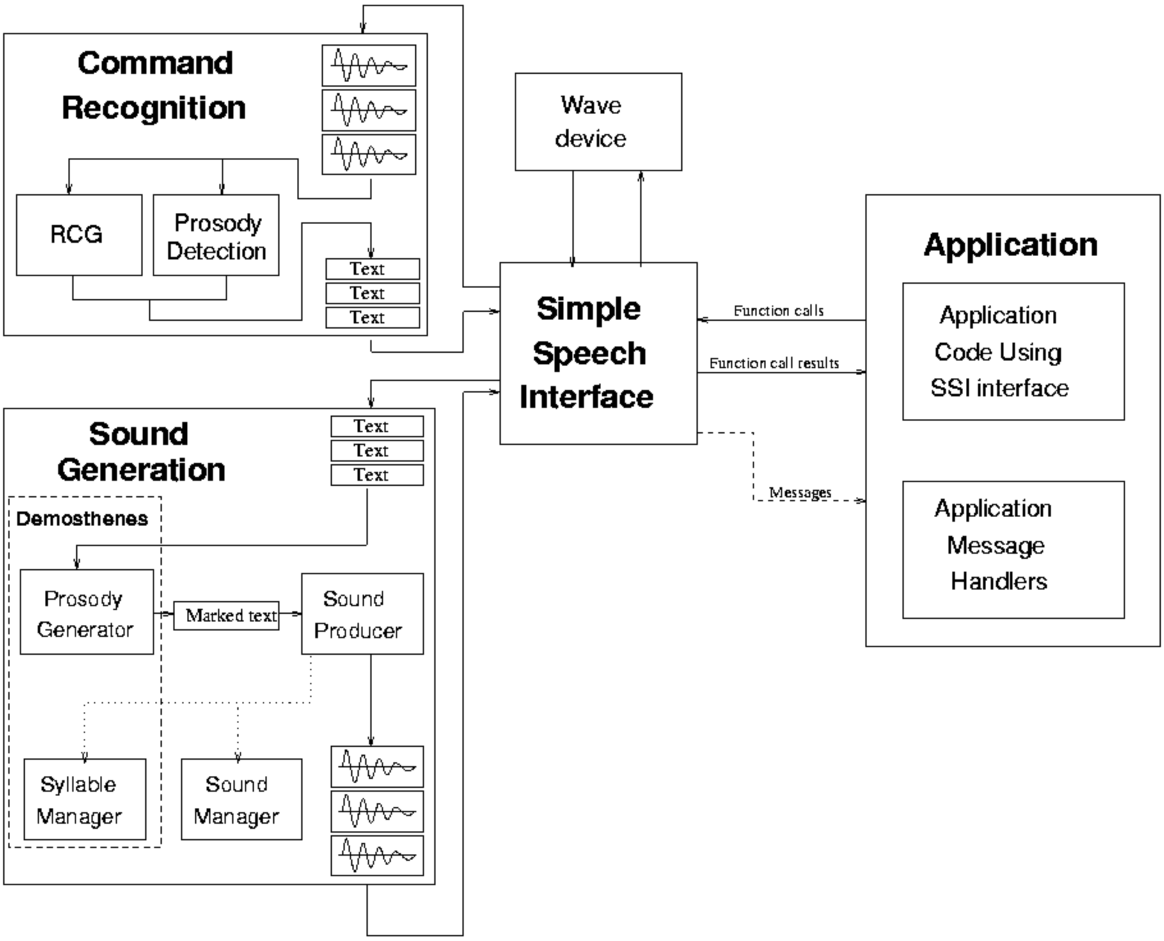

Figure 1. *Structure of the DIALOG Speech Interface System*

5. Applications

Several applications intended for blind people are currently developed, e. g. the Emacspeak system [12] or audio www browsers pwWebBrowser [10] and the IBM Home Page Reader [11]. Their main purpose is to read the textual contents of the screen more effectively than a screen-reader software is able to do. Audio web browsers usually use options provided by W3C consortium as a part of its Web Accessibility Initiative specification. Frequently used features are reading 'ALT' attribute for images, 'SUMMARY' attribute for tables or using extensions defined in CSS specification. However, such systems have some drawbacks: they do not use spoken commands, they do not involve a user model and above all they do not include any dialogue processing component. In fact, they cannot be called dialogue systems. Applications described below should overcome these insufficiencies. Actually two applications are briefly presented – the programming system DIALOG and the speech oriented hypertext system AUDIS.

The programming system DIALOG [4] is developed with the intention to provide blind user programmers (blind people do write computer programs!) with a programming language and integrated development environment adapted to their special needs. The system is designed to suppress syntactical and also some semantical errors as much as possible. In this way blind users are able to orient themselves as quickly as possible in the source code.

All the communication of the user and the system is accomplished via dialogue. A dialogue in a standard form supports source code generating, editing, testing, searching, debugging etc. The conversation is fully supported by speech input and output. Non-speech sounds are used to speed up some often used actions and subdialogues non-speech sounds are used. Source code and data have the consistent structure. Basic entities of the system are *objects*. Objects represent user-defined data or commands of the provided programming language, such as conditions, cycles, calling external functions etc. Objects are organized into a tree. To each object a set of *attributes* is assigned depending on the command which is represented by the object. DIALOG objects are designed to facilitate implementation of the dialogue processing module that prevents blind users from making errors. Each object has a name. DIALOG source code tree is naturally represented in its computer form. However, whenever the textual transcription of an object tree is needed, we use the XML data format. In what follows a classical simple program “Hello, world” written in DIALOG is presented:

```
<?xml version="1.0"?>
<!DOCTYPE PROGRAM SYSTEM "dialogprogram.dtd">
<PROGRAM Name="Hello, World">
     <FUNCTION Name="main">
          <CALL Function="TELL" Arguments="'Hello, world'"/>
     </FUNCTION>
</PROGRAM>
```

For more detailed information see *http://www.fi.muni.cz/lsd/dialog/*.

The hypertext browser AUDIS ([1], [5]) is being developed since 1998. It is primarily intended for visually impaired students. One of the most serious problems for the students with sight impairment is to acquire studying materials in the appropriate form. AUDIS should allow these students to access the studying materials quickly and with respect to their needs.

AUDIS cooperates with the Interface module to perform speech communication. It uses all of its capabilities (voice commands, speech synthesis output, earcons and environmental sounds) to provide the blind users with structured textual information quickly and clearly. Special attention is paid to facilitate blind user orientation in the hypertext document structure. AUDIS also supports special graphical output for people that are not totally blind. The system creates the hypertext structure of the document that supports blind user orientation in the document. Document is divided into chapters, sections and subsections. These components are organized into a tree. Each component contains a short summary at the beginning. All chapters, sections and subsections are linked to the previous and next item at the corresponding level of the tree. Following these links user can traverse the document linearly. Moreover, information about the current chapter, section and subsection is accessible at any time. Currently, AUDIS documents have to be manually prepared.

HTML has been chosen as the AUDIS document format. The format is relatively simple, widely used and has an open standard. Thanks to this conception, extension of AUDIS functionality (which is planned in the future) to the World Wide Web browser will be quite straightforward.

Conclusions and Future Work

Developing the user interfaces and spoken dialogue systems oriented to visually impaired people brings specific problems that can be partially solved by adapting of the dialogue strategy. In the future work we would like to perform further testing of the possible extensions and modification of the method described in the paper (especially by WOZ simulation) to verify and enhance the effectiveness of the method. The development of the communication module is directed towards enabling more freedom in communication. This should be achieved by implementation of the language syntactic parser and semantic analyser [8].

Acknowledgement

The authors are grateful to K. Pala for reading the draft of the paper and valuable comments. The research has been partially supported by Czech Ministry of Education under the Grant VS97028 and by Grant Agency of the Czech Republic under the Grant 201/99/1248.

References

[1] L. Bártek, R. Batůšek, M. Fikera, I. Kopeček: *Using Dialog Programming for Developing the Speech Oriented Hypertext AUDIS;* Proceedings of TSD'98, 1998, pp. 361-364.

[2] Darvishi, A.: *World Wide Web Access for Blind People: Problems, Available Solutions and an Approach for Using Environmental Sounds;* Proceedings of ICCHP'96, pp. 369-373.

[3] Kahlish, T.: *Improving Access to Hypertext Based Study Material for the Blind;* Proceedings of ICCHP'96, Linz, 1996, pp. 229-236.

[4] Kopeček, I.: *Dialog Based Programming;* Proceedings of the First Workshop on Text, Speech and Dialogue - TSD'98, 1998, pp. 407-412.

[5] Kopeček, I.: *The Architecture of the Blind User Oriented Hypertext AUDIS;* in Proceedings of ICCHP'98, Wien-Budapest, September 1998, pp. 215-223.

[6] Kopeček, I.: *Syllable Based Speech Synthesis;* Proceedings of the 2nd International Workshop Speech and Computer - SPECOM'97, Cluj-Napoca, 1997, pp. 161-165.

[7] Kopeček, I: *Syllable Based Approach to Automatic Prosody Detection; Applications for Dialogue Systems;* Proceedings of the Workshop on Dialogue and Prosody Eindhoven, 1999.

[8] Pala, K.: *Semantic Annotation of (Czech) Corpus Texts;* Proceedings of the Workshop on Text, Speech and Dialogue - TSD'99, 1999.

[9] Stevens, R.D., Wright, P.C., Edwards, A.D.N., Brewster, S.A. (1996) *An Audio Glance at Syntactic Structure Based on Spoken Form;* Proceedings of ICCHP'96 1996, pp. 627-635.

[10] The Productivity Works pwWebSpeak: http://www.prodworks.com/productsindex.htm

[11] IBM Home Page Reader: http://at.rsb.org.au/products/ibmhpr.htm

[12] Raman T. V.: *Emacspeak – A Speech Interface*; Electronic Proceedings of CHI'96, http://www.acm.org/turing/sigs/sigchi/chi96/proceedings/papers/Raman/paper.html, 1996