

Open source technologies for visually impaired people

Patrizia Boccacci¹, Veronica Carrega¹, and Gabriella Dodero²

1 DISI, University of Genova, Via Dodecaneso 35, Genova, Italy
boccacci@disi.unige.it

2 Free University of Bozen, Dominikanerplatz 3, Bozen, Italy
gabriella.dodero@unibz.it

Abstract. We describe two open source applications which we have experienced as very useful aids for the integration of people suffering from visual impairments, from hypovision to actual blindness. The first application is based on speech synthesis and has been experienced by disabled university students. The second experience is oriented to schoolchildren with low residual vision, and it provides their educators and parents with easy to use tools for image manipulation, especially designed for exploiting residual visual abilities.

1 Introduction

The increasing interest in the promotion of education for visually impaired students, at any level, has correspondingly increased also the number of blind university students, who successfully graduate in all subjects. At the same time, EU and Italian legislation are enforcing the adoption of accessible technologies to help disabled citizen in their studies, work and social life. In the same countries there have been also many actions in support of open source technology adoption within public administrations. Yet today most software to be used as computer based aid for disabled citizens is proprietary, and it is often acquired with public funds, by schools, universities and by National Health Services.

The possibility to use open source aids for disabled people has been investigated by the authors, having in mind the specific needs of a blind student, who enrolled in Fall 2005 as BSc student in Physics. Desktop Linux distributions, which already integrate speech synthesis tools, could not be installed in student labs because data acquisition hardware, used for performing physics experiments, was not supported under such distributions. As a consequence, a thorough study of what OS speech synthesis software was available was undertaken, and finally a lab workstation was configured, running open source software, and software available for free to non-profit organizations[1]. This workstation provides speech synthesis in Italian and in English under Linux, with easy keyboard commands to switch between the two languages. No Braille device needed to be installed, only earphones. While speech synthesis may substitute reading, information conveyed by images cannot be perceived by blind students. Students with low residual vision may still view images

by means of optical devices (magnifiers) enlarging them, but such tools are heavy and expensive, and a student can seldom use them both at school and at home. For this reason, we investigated the features of open source image manipulation tools, and developed an easy to use interface, especially conceived for the educators and parents of visually impaired children. The image manipulation tool performs contour extraction and "thickens" the lines so that images can be clearly perceived. In the rest of the paper we shall describe how we use open source speech synthesis tools, and then the image processing tools which we have developed. We conclude with some comments on this experience, and on its relevance for the open source community.

2 Speech synthesis

Speech synthesis is the process of generating a human voice by means of electronic devices connected to a computer. It consists on two main steps. The first step is text analysis, where the input text is transcribed in a phonetic representation. The second step is the generation of sounds (waves), where audible output is made with a combination of phonetic and prosody information. The process is sketched in Figure 1. The software module implementing the first phase is often called NLP (Natural Language Processor), while the second is called DSP (Digital Signal Processor).

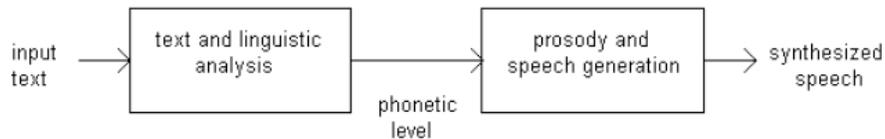


Fig. 1 Organization of a speech synthesis suite

NLP contains translations from letters to sound, and prosody generation, but also a morphological and syntax analyzer. Extraction of a syntax tree from the sentence is in fact extremely useful to achieve good phonetical and prosody translations.

Most DSPs use a voice database, that is a collection of registered speech fragments. Elementary components of such database are sounds (a vowel or a consonant) or better, diphones, which include the second part of a sound and the first part of the next sound (for example the transition between a vowel and a consonant). Starting from phonemes and prosody, the DSP outputs a sequence of segments, which are adapted to the required prosody, eliminating discontinuities; finally the resulting flow is synthesized and the voice is output[2]. Several software tools are available under Linux, but most of them were of no use for us, since they support only English, or provide marginal support to other languages. The main cause is the length and cost of creating a database of sounds containing all the relevant diphones for each language, and the difficulty in expressing and tuning the rules for transforming texts into phonemes.

One of the most promising speech synthesis systems is MBROLA[3], which has a database of phonemes for about thirty languages, including Italian[4].

Unfortunately, the licensing policy of such project is quite restrictive: MBROLA can be distributed only as a precompiled binary, and only non-military and non-commercial applications may use its databases of phonemes. In other words, it is not an open source product, even if it can be available for free, and several open source tools are compatible with it. The BLINUX website[5] provides many (but not all) open source resources for visually impaired users. It collects a huge amount of packages of several kinds, from Braille device drivers to more complex systems like speech synthesis modules. The selection of suitable packages within BLINUX is not straightforward, and of course, care must be taken in order to avoid old systems, no longer compatible with recent kernel versions, or without any more support groups of active developers and users. For speech synthesis, this problem is especially important since there is no single tool solving the problem, and we need a suite of interoperable applications. The next subsections describe our choices.

2.1 Festival

Festival[6] has been developed at Edinburgh University, as a research tool in speech synthesis. It includes a complete text-to-speech conversion module, which can be invoked by a shell, by a command interpreter, by Java or by an EMACS interface. The system is written in C++ and uses the Edinburgh Speech Tools Library for low level operations. It can be used as a standalone tool, or as a development environment for further speech synthesis tools. Both Festival and the Speech Tools are distributed without licensing restrictions, for both commercial and non commercial use. Festival supports several languages, including Italian, by installing additional files. Italian phonetics has been developed at the Padua Laboratories of the National Research Council; such database can be used together with the MBROLA speech synthesizer (as we did), and provides different voices for the language, including both male and female voices.

2.2 Speech Dispatcher

Since the lab will be used in various scientific related tasks, it is important to integrate speech synthesis within an environment like EMACS. A system capable of connecting Festival to EMACS is Speechd-el and Speechd. The core of the system is Speech Dispatcher [7] (called also Speechd), which provides a device independent level for speech synthesis, across a stable and well documented interface. Speech Dispatcher is implemented as a client-server system: applications invoking speech synthesis functions are clients, which establish a TCP connection to Speechd through SSIP (Speech Synthesis Independent Protocol). Libraries for interfacing to many languages are available (C, Python, Lisp, Common Lisp). Speech Dispatcher translates requests from client applications into commands to the synthesizer, handles message priority, keeps trace of requests logs, and provides the so-called sound icons functionality. Sound icons are sound sequences informing the blind user about the execution of particular actions, such as opening, saving and closing a file.

Once a client application has established a connection, it invokes functions like say(), stop(), pause(). Speech Dispatcher parses such commands, reads the text to be pronounced, and queues it in accordance with requests priority. Then it selects when, by which synthesizer, and with which parameters such a text shall be output. In this way, applications need not deal with low level details in speech synthesis.

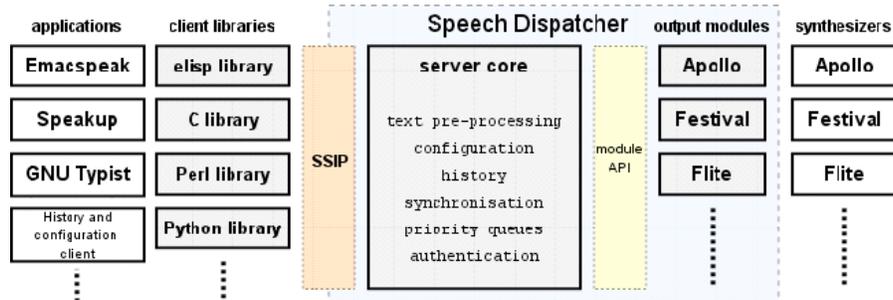


Fig. 2. Speech Dispatcher architecture and interfacing

Figure 2 shows the structure and links between Speech Dispatcher and applications on one hand, speech synthesis systems on the other hand. In fact, Speech Dispatcher interfaces with many speech synthesizers, including Festival. Speech Dispatcher interfaces with EMACS by means of the libraries Festival-freebsoft-utils, and through the Speechd-el client. The latter is a voice output system for EMACS which provides an Emacs Lisp library in order to make available a subset of Speech Dispatcher functions. This makes EMACS totally accessible to blind users.

2.3 Gnopernicus

Gnopernicus[8] is a system especially designed for hypovision; it allows people with low residual vision, and even blind Braille readers, to use the GNOME desktop and its applications. It provides Braille output, screen magnification, screen reading; it may interface to synthesizers like Festival. It provides an Assistive Technology Support menu, which can be used to configure system startup and activate Gnopernicus at login, making the blind user totally free from external help.

3 Image Processing

By discussing with therapists, educators and parents of visually impaired children, we realized that simple magnification of an image contained in a textbook may not be sufficient to provide relevant information to disabled pupils. In fact, images often contain details, irrelevant to the intended educational purpose. Such details are automatically discarded by normal children, while for visually impaired children their selection wastes much effort. Also the use of many bright colors, intended to capture the attention of normal children, can be confusing for a disabled pupil, who would most appreciate a line drawing with contrasting background and no further details. The most severely impaired people, who cannot perceive an image with color

or greyscale details, often may see such a simple sketch drawn with a thick line. There exist already several open source tools, which may perform suitable image processing and may make digital images better perceivable to hyposeeing people. However, none of such tools has been designed for being used in such a specific domain, and the desired functionalities should be sought for within menus and options, discouraging parents and educators from using them. To help non specialists to convert, automatically, color or greyscale digital images into sketches visible by hyposeeing people, we developed a plugin which extends the features of a well known open source tool, ImageJ. ImageJ is a public domain Java image processing program that runs, either as an online applet or as a downloadable application[9]. ImageJ has an open architecture that provides extensibility via Java plugins. Custom acquisition, analysis and processing plugins can be developed using an editor and Java compiler. Our plugin provides a simplified interface which skips all ImageJ menus containing unnecessary features. The user, through an accessible menu, can set her/his personal profile, selecting between three degrees of resolution (low, medium,high), three thicknesses of line (thick, medium, fine) and color pairs for background and lines (black/white, yellow/blue etc). The plugin converts color images into grayscale, and then applies a standard algorithm[10] of edge detection, using threshold computed from user profile data. Figure 3 shows an original image, and the resulting image after conversion with our plugin: only essential details are kept, and the resulting image is visible to seriously hyposeeing people.



Fig. 3. Original and converted image of a glass bottle

4. Conclusions

Aids for disabled people represent an application domain which seems especially suited for the open source paradigm, since the unavoidable need for adaptation to actual degree of disability is best satisfied by open source software. In practice, this has seldom occurred till now, probably for lack of awareness about the existence of

viable open source disability aids. Our experience has shown that existing open source tools may represent good disability aids. Of course, proprietary software aids exist as well, with comparable or higher quality. However, the quality of open source solutions is already sufficient for many users. Someone objected to using MBROLA as a component of our speech synthesis solution, since it cannot be considered open source software. This however may be a temporary solution, since an open source alternative, SMS[11], is almost ready for being released. Use of such software together with Festival shall allow us to rely on open source software only.

The fact that speech synthesis open source software exists does not mean that it is trivial to run it and that our installation shall represent a solution for all hyposeeing problems. From users perspective, knowledge of Braille, presence of residual vision and personal preferences may suggest a completely different approach with respect to ours. And considering Linux installations, care must be taken to select audio peripherals that are already supported by recent drivers, under the preferred distribution. Similar considerations hold for image processing tools: existing software has to be made available through an accessible and simplified interface, in order to be really usable for our purposes. In both cases, we experienced the interaction with developers and users communities, who gave useful feedbacks on how to solve configuration and interoperability problems, and how to modify or extend the available code, whenever needed.

Acknowledgement

The authors are indebted to Silvia Dini from Istituto Chiossone, Genova, for fruitful discussions and feedbacks about possible aids for visually impaired people.

References

- [1] V.Carrega, Open Source tools for visually impaired people, Master Thesis, Università di Genova (2006), in Italian.
- [2] F. Felletti, Text-to-speech technologies applied to website navigation, Master Thesis, Università di Ferrara (2004), in Italian.
- [3] T.Dutoit, V.Pagel, N.Pierret, F.Bataille, O.Van Der Vrecken The MBROLA Project: Towards a Set of High-Quality Speech Synthesizers Free of Use for Non-Commercial Purposes, in: Proc. ICSLP'96, Philadelphia, vol. 3 (1996)pp. 1393-1396.
- [4] P.Cosi, R.Gretter and F.Tesser, Festival speaks Italian, in: Proceedings of GFS2000, Padova (2000), in Italian.
- [5] Blind + Linux = BLINUX, (February 23, 2007) <http://leb.net/blinux/>
- [6] R.A.J. Clark, K. Richmond, and S. King, Festival 2 - build your own general purpose unit selection speech synthesiser, in: Proc. 5th ISCA workshop on speech synthesis (2004).
- [7] Speech Dispatcher, (February 23, 2007) <http://www.freebsoft.org/speechd>
- [8] Introducing Gnopernicus, (February 23, 2007) <http://developer.gnome.org/projects/gap/AT/Gnopernicus/>
- [9] ImageJ, Image Processing and Analysis in Java, (February 23, 2007) <http://rsb.info.nih.gov/ij/>
- [10] J. Canny, A Computational Approach to Edge Detection, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 8, no. 6 (1986) pp. 679-698.
- [11] G.Sommavilla, C.Drioli, P.Cosi, G.Tisato, SMS-FESTIVAL A new framework for open source speech synthesis, Proc. AISV conference, Trento, (2006).