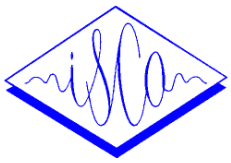
ISCA Archive
http://www.isca-speech.org/archive



The ESCA Workshop
on Speech Synthesis
Autrans, France
September 25-28, 1990

IMPLEMENTATIONAL ASPECTS AND THE DEVELOPMENT SYSTEM OF THE MULTIVOX TEXT-TO-SPEECH CONVERTER

Géza Németh , Géza Gordos , Gábor Olaszy

Speech Research Laboratory of the Technical University of Budapest
Budapest, Stoczek u.2. Hungary 1111
Phonetics Laboratory of the Hungarian Academy of Sciences
Budapest, P.O. BOX.19. Hungary 1014

ABSTRACT

MULTIVOX is a general purpose, multi-lingual, real-time text-to-speech (TTS) system for IBM PC and compatible computers in the following languages: Hungarian, German, Finnish, Italian and Esperanto. French, Spanish and Portuguese versions are under development. This system has been developed as a joint effort between the Speech Research Laboratory of the Technical University of Budapest (TUB) and the Phonetics Laboratory of the Hungarian Academy of Sciences (HAS). In this paper its implementatitonal aspects and a short description of its development system will be covered while its phonetic aspects are described in a separate paper [Olaszy et al., 1990].

1. INTRODUCTION

Since the early eighties there has been continuous cooperation between the TUB and the HAS in the area of formant synthesis [Czapp et al., 1988 and Olaszy, 1989). In the HAS most of the phonetic, linguistic basic research was conducted, while the TUB provided mainly the implementational, algorithmic, hardware and computing expertise. The latest result of these activities is the MULTIVOX system.

2. IMPLEMENTATIONAL ASPECTS

2.1. General considerations

Our goal was to develop a formant synthesis environment, which is based on a simple hardware platform and makes it possible to integrate research, development and application programming into a single workstation. The main research focus was on multi-lingual TTS. The commercial version had to use the same hardware and basically the same software as the development one, so that time consuming recoding and hardware modifications could be avoided. We hope, that this approach may result in a good compromise between quality and simplicity. Real-time operation was a must from the beginning. As regards the target languages a step-by-step approach was chosen. Starting with our mother tongue, Hungarian, continuing with the Finno-Ugric Finnish, then the languages of geographically and culturally easily accessible countries were studied. English TTS was excluded from the research program as it seemed to be a too large task with less promising results in the foreseeable future.

2.2. Hardware

As regards the hardware basis for the implementation of the synthesis

algorithms, a choice had to be made between a digital signal processor (DSP) and a ready-made, commercially available, freely programmable formant synthesis chip. Although the former provides maximum flexibility, still the latter was chosen, because it offers much advantage in lower chip count and in case of possible low power applications. The fully CMOS PCF 8200 from Philips contains digital noise and periodic signal sources, five programmable formant filters, an 11 bit D/A converter, interpolation and uP interfacing logic with some other extra features, too.

This choice made it possible that a simple, easily installable output device could be designed. It is of the size of a small hand-held radio, and contains the synthesizer chip, the audio amplifier, some interfacing logic and the speakers only. It can be controlled through the industry standard Centronics interface. The device , called SPEAKTRONICS, is the same for all languages. It can be connected to the host without opening its cover. An audio line output was also included for the purposes of tape-recording or external amplification. Headphone output and a volume knob serve the possibility of quiet operation for the surroundings.

The IBM PC and compatibles were chosen as host. The reason is rather straightforward: this computer family is applied in a multitude of application areas. The output of research was intended for as many end-users as possible.

2.3. Software

The choice of the host was that of the operating system (OS), too. On PC's and compatibles MS DOS is the most often used OS. A relatively late version (3.3 or later) had to be applied as the earlier ones didn't provide proper memory allocation and task switching procedures. Thanks to the rather simple structure of the system, no special arrangements had to be made to reach real-time operation on a standard 4.77MHz PC. As the high end of the PC family works with as high clock frequencies as 30-40MHz special care had to be taken of the correct interface handling, that both high and low end computers could properly communicate with the SPEAKTRONICS box. For the sake of easy maintenance and structured programming, most of the programs were written in high level languages, MS Pascal and C. Only the hardware dependent and the interrupt handling routines were written in assembly. The memory requirement is appr. 100-200kbytes, depending on the language. If it is necessary it can be made smaller by sacrificing the large size (max. 250 characters) of a sentence that can be handled as a single unit.

The application user interface posed a rather difficult problem because speech output is not a standard device of the OS as yet. Offering the end-user system as an add-on library for different programming languages and compilers could have easily created a mess. Our solution of this problem was that the end-user TTS system operates as a Terminate and Stay Resident (TSR) program that can be driven through a software interrupt. This interface layer is quite well defined and often used, so most programming languages provide some tools for its application. If it is necessary, assembly language interface drivers can be easily developed.

From the application programmer's point of view, the TTS system operates very similarly to a printer. The input is a character string, while the output is the acoustic phenomena and a word return value. Each bit of the return value signals a certain type of error. As the bits are set to one only if some error

occurred , error detection and handling can be easily accomplished. The programmable options (more than 15) of the system can be set also in a printer-like manner by so-called escape sequences.

3. THE DEVELOPMENT SYSTEM

3.1. Development system components

The interactive development system consists of the following components:

- -SPEAKTRONICS output device
- -extended MULTIVOX system with monitoring capabilities
- -screen editor for the different databases
- -statistical routines for the databases

The seemingly close link between the end-user and the development system makes the development work really effective. The acoustic characteristics of the output device can be taken into account during the development phase , so no later adjustment is needed. The same advantage holds for the extended software and the end-user system. The source code of most of their modules is actually the same, only compile time conditional options determine, whether end-user or development version will be compiled. The different software modules can be accessed as popup menus from a main menu. Shortcuts between corresponding modules are also available.

3.2. Extended MULTIVOX system

The extended MULTIVOX system operates as a sort of demo program. It occupies the full computer screen. The upper half of the screen contains a simple text editor where any test character sequence can be input just as if it would be for the end-user system. After the proper text or sound sequence (e.g. baba or papa) has been prepared, it can be sent to the TTS system with a single keystroke. During the conversion process the corresponding phoneme codes are displayed in the lower half of the screen. Optionally it can also be displayed, which lines of the saw-teeth like grapheme -> phoneme conversion chart are being used. This option is especially useful in case of those languages where this table contains a large number of rules (e.g. German 1180). At the end of the conversion process we can hear the synthesized text from the output device.

3.3. Database editors

The only way to judge the correct operation of the TTS system and to experiment with the synthesizer chip is to listen to it and to view and modify the formant synthesizer control codes themselves accordingly. An alphanumeric full-screen editor is provided for these purposes. The parameters are displayed in natural units (i.e. pitch, formant frequences and bandwidth in Hz, frame length in ms, etc.). The frame number within the given phrase, the corresponding ABU inventory code number , the sign of the phoneme to be realized and a code corresponding to intonation changes are also displayed. These parameters give a complete overview for the user of the systems operation in the phoneme code to control code phase. Frames can be freely deleted, changed or inserted so manual experimentation can be easily carried out. With the addition of a vocabulary editor, a limited vocabulary word preparation system can also be created [Németh et al., 1989].

If some regular sound errors occur in the TTS conversion process, a change has to be made either in the low level acoustic building unit (ABU) database or in the rule system that describes the phoneme -> ABU relationship. Both can be easily changed with the corresponding editor. It is worth mentioning, that the raw acoustic database occupies only 1275bytes, while the rule system needs appr. 10kbytes for German. A universal voice character editor is also under development which will serve for transforming the ABU inventory between different voice types. The grapheme -> phoneme conversion chart can be edited by any word processor. The size of this table is appr. 20kbytes for German.

3.4. Statistical routines

As the ABUs themselves are meaningless units, they can be used only by the rule system, describing the phoneme -> control code relationship. If we want to improve the realization of sound combinations, we have three ways:

(i) -use another ABU in the given rule e.g. (2,12,2) -> (3,12,2)
(ii) -change an ABU in the rule e.g. F2 increased in ABU 12 with 2 steps
(iii) -create a new ABU

(i) Changing one ABU code to another can be done without any problem with the rule system editor.
(ii) Care should be taken however if we want to change something in an ABU, because one ABU can be used in several rules. If we change it without proper consideration, we can improve one sound combination by spoiling several others. That's why one can list the rules in which a given ABU is applied, on the screen.
(iii) If it is used in several rules it is advisable to create a new ABU and use that one for the improvement. Another routine helps this process, which can list the unused ABU code numbers.

4. NEW LANGUAGE DEVELOPMENT

As it could be seen from the above description, no intermediate language is used for multi-lingual synthesis. Only the hardware and the software structure are the same. During the development of the currently available five languages it was found, however, that appr. 30-50% of the rule system and the ABU inventory differ only during the conversion from one language to another. It is important to note, that the sound set, the grapheme -> phoneme chart and the intonation routines are uniquely created for each language. The unified simple structure ensures easy maintenance, and the problem-free introduction of mother-tongue non-expert persons into the operation of the system.

REFERENCES

Czapp, Gordos, Németh, Olaszy, Tihanyi (1988),"An integrated approach to text-to-speech and fixed vocabulary formany synthesis", Proceedings of the VDE-ITG conference on digital speech processing, Bad Nauheim,213-216.

Németh et al.(1989),"Embedding speech synthesis into applications",Proceedings of the Speech Research '89 Conference,Budapest,285-288.

Olaszy G.(1989),"Speech synthesis in Hungary from the beginnings up to 1989", Proceedings of the Speech Research '89 Conference,Budapest,289-292.

Olaszy G.et al.(1990),"Phonetic aspects of the MULTIVOX text-to-speech system", in the same volume