Adaptation of the AhoTTS Text to Speech System to PDA Platfoms

Jon Sanchez, Iker Luengo, Eva Navas, Inma Hernaez

Department of Electronics and Telecommunications
University of the Basque Country. Spain.
ion,ikerl,eva,inma@bips.bi.ehu.es

Abstract

This paper presents the work carried on to adapt a Basque language Text-To-Speech (TTS) system into a mobile device of limited resources. The aim is to make possible the use of the AhoTTS conversion system of the UPV/EHU´s Aholab group, in a Personal Digital Assistant (PDA), and to test the system performance in several aspects, such as sound sample generation times. The selected PDA is a Pocket PC under the Windows CE operating system. The converter has been compiled in a library which provides an Application Programming Interface (API) to the applications. Some applications that use the created API are also described.

1. Introduction

Speech Technology is getting large use in our environment. Speech is being increasingly used in human-machine interfaces, since it is a natural way to communicate for the human being. To get a completely oral interface, two basic systems are required: a speech recognition module, which will get voice messages and interpret them, and a speech synthesis module, which will create a voice signal that people can understand.

In the other hand, wireless technologies are creating the background for a wide variety of portable devices. Nowadays, there are several devices that, having a Pc-like architecture, use very different software and provide with different mobile resources.

Personal Digital Assistants (PDAs) are experiencing a certain growth during the last years, and they are not longer little more than plain agendas, but complete pocket computers, providing Internet connection and a large amount of applications.

One of the main drawbacks of these devices is the limited input/output interface. They don't have usual hardware elements like keyboard or mouse, and the interaction takes place in a small tactile screen, where it is not comfortable to read or write. Due to these limitations, speech appears to be a natural and efficient communication way: reading can be substituted with a Text to Speech converter, and writing can be performed with Speech Recognition software.

By this work, Basque language text to speech synthesis capabilities are added into a PDA, improving its interface. Since Basque is an endangered language, used by just about a million people in Europe and America, it is very important to develop technologic support for the Basque speakers, so they can continue using it normally in every aspect of the life.

2. AhoTTS conversion system: description

The AhoTTS system [1] developed by the Aholab group in the University of the Basque Country is a modular text to speech synthesis system. It has a multithread and multilingual architecture, and every module has been developed for both Basque and Spanish languages.

The TTS is structured in two main blocks: the linguistic processing module and the synthesis engine, as in Figure 1. The fist one generates a list of sounds, according to the Basque SAMPA code [2], which consist on the phonetic transcription of the expanded text, together with prosodic information for each sound. The synthesis engine gets this information to produce the appropriate sounds, by selecting units and then concatenating them [3] and performing a processing to reduce the distortion that appears due to the concatenating process [4].

3. TTS Adaptation

3.1. System specifications

The converter modules are C/C++ written applications, ported to several platforms and operating systems. I. e., for the linux operating system, the converter can be used in Intel and Sun

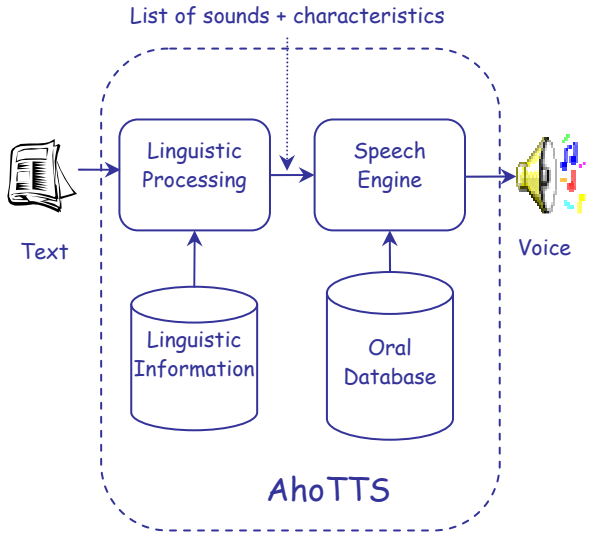

Figure 1: Structure of the AhoTTS conversion system

machines, and there is also a version for the Windows operating system. There is also a network version, available through the Internet[1].

One of the main advantages of the system is its portability. There is a single version of the code that, with the necessary libraries and macros for each platform, can be used to develop the software under different compilers. For the PDA, a programming station has been installed on a Windows PC machine.

Regarding the hardware requirements, it is necessary to have a sound system available on the PDA in order to use voice. It is also necessary to have enough processing power to generate and play audio samples. The model used to carry out this work was a HP iPAQ h2210, with 400MHz processor speed, 64Mb RAM and Pocket PC 2003 Operating System.

3.2. Migration

Microsoft Windows CE [5] is a real-time modular operating system, with implementations for different 32 bit architectures. In Handheld systems, the most used implementation is Pocket Pc 2003[2]. It uses the Microsoft eMbedded Visual C++ 4.0 developing software, with an appropriate Pocket PC 2003 SDK.

The eMbedded Visual C++ 4.0 compiler has got some limitations: it is not provided with every C standard library or function, in order to get smaller and faster applications to run in the limited environment of a PDA. In detail, the AhoTTS system requires the ability to use complex numbers, and makes use of the 'iostream' library functions, which were not implemented. To get this functionality, Dinkumware's Dinkum UnAbridged Library for VC++ [6] was used.

Using this platform, an Application Programming Interface (API) has been developed, in two different ways: as a static library, and as a dynamic library. Both of them can be used to link programs with voice capabilities.

3.3. System performance

The designed system is aimed to perform real-time tasks, so the time taken to generate a sound must be shorter than the sound duration itself. Since the voice will be generated with a 8KHz sample rate, every sample must be generated in a 125μs time, in the worst cases.

Three different tests have been carried out: the first one to compare the performance of the different versions of the API with the counterpart in a Pc, the second one aimed to evaluate the performance of the software in different PDAs, and the third one comparing the performance with different database loading methods.

3.3.1. Library linking method

The API has been developed as a statically or dynamically linked library. This first test compares the performance of both linking possibilities with the counterpart Pc system. Table 1 shows the result of the first test. The PDA system was the HP iPAQ h2210, and the PC system was a Windows platform with 512 Mb RAM and 1300 MHz processor speed.

The system configuration, for every test, involves a male voice in Basque, the MBROLA synthesis engine and database preload configuration. As expected, the performance of the PDA system is not as good as that of the PC machine, but it is still enough to get real time response. It is also true that the dynamically linked system performs better than the static one.

Text length	Static Library mean time per sample	Dynamic Library mean time per sample	PC mean time per sample
100 chars	11 μs	10μs	0,8μs
200 chars	17μs	11μs	1μs
500 chars	29μs	21μs	1μs
1300 chars	-[3]	21μs	1μs
3000 chars	-[3]	21μs	1μs

Table 1: TTS Performance in PDA and PC systems.

3.3.2. Real-time capabilities in different hardware

The second test played a 3000 character Basque test in five different PDAs. Their characteristics are shown in table 2. The text was used in a single thread conversion task, and then in five parallel threads, to measure the performance of the system in both multithread and single thread requests. The dynamically linked library has been used, with a MBROLA conversion engine and a database preload configuration.

PDA	RAM (Mb)	ROM (Mb)	Processor (MHz.)
HP iPAQ 2210	64	32	400
HP iPAQ 6300	64	64	168
Dell Axim X50	64	32	520
Acer N35	64	64	266
Toshiba PXA263	128	32	400

Table 2: PDAs which carried out the real-time performance tests, and their characteristics.

The results of this second test appear in images 2, 3, 4, 5 and 6. Horizontal lines show the mean sample generation times for the single thread task (white line) and the multi-thread test (black line). Also, the vertical bars represent the mean sample generation time for each thread of the multi-thread test.

[1] http://bips.bi.ehu.es/tts/tts_en.html

[2] And other related ones, like the recently appeared Pocket PC 2005.

[3] The static library was not able to convert texts larger than 1000 characters.

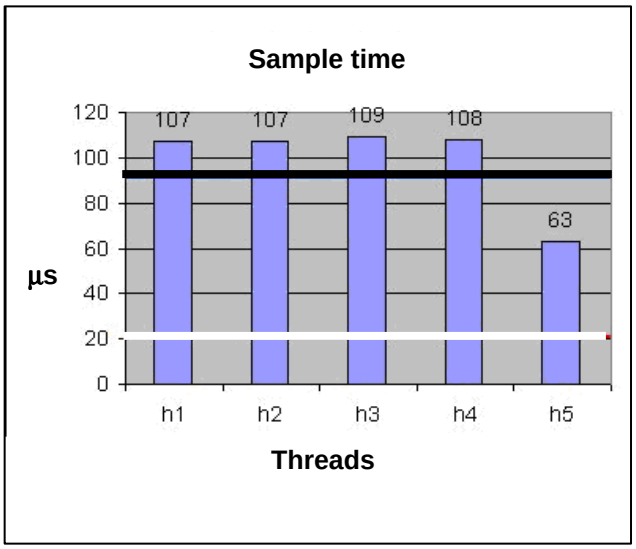

Image 2: Result of the performance test in a HP iPAQ h2210 PDA.

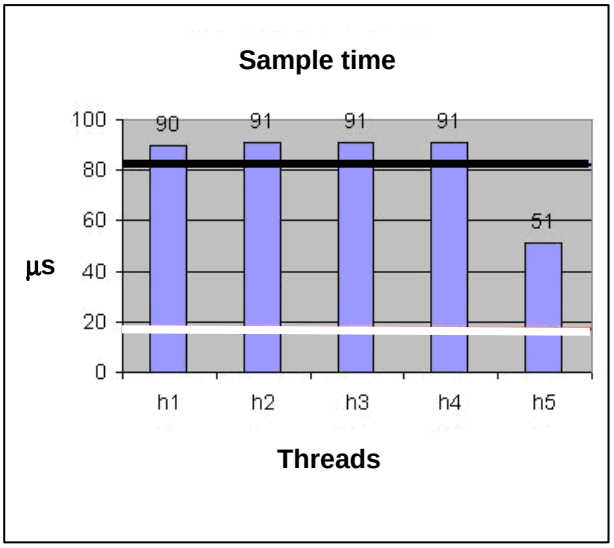

Image 4: Result of the performance test in a DELL Axim X50 PDA.

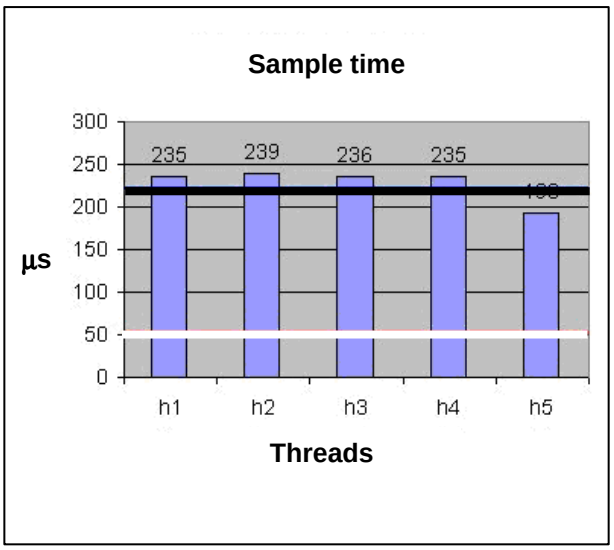

Image 3: Result of the performance test in a HP iPAQ 6300 PDA.

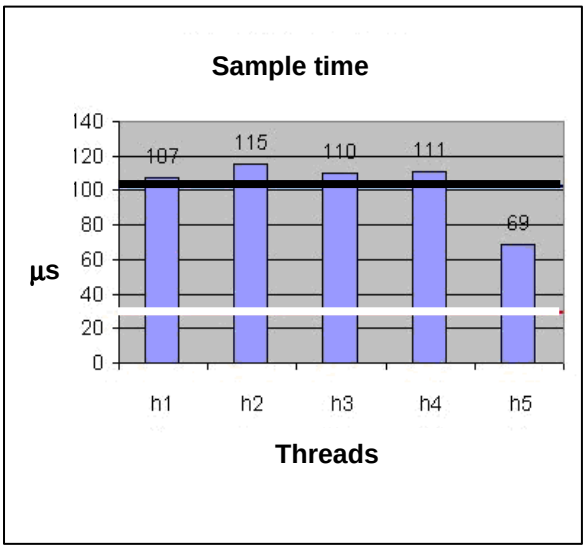

Image 5: Result of the performance test in a Acer N35 PDA.

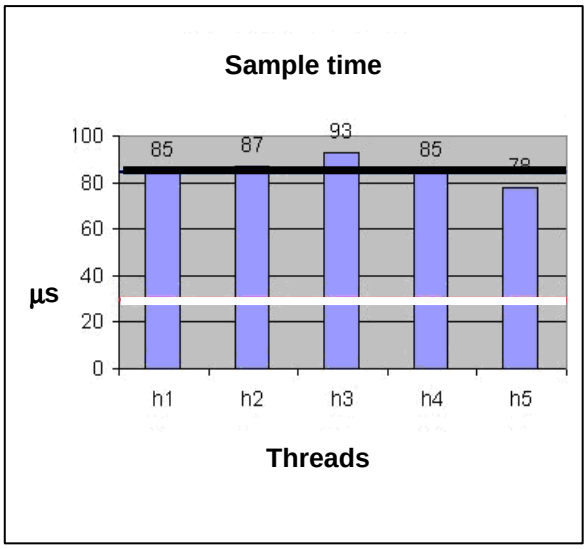

Image 6: Result of the performance test in a Toshiba PXA 263 PDA.

In short, the single thread test got shorter times than the maximum allowed sample time in every PDA tested, while the multi-thread test did not succeed en the HP iPAQ 6300, because the mean time used to generate a sample is larger than the sample time. According to table 2, this PDA has got the slowest processor, being the other characteristics similar. So, we can conclude that 168MHz processor speed is not enough to perform a multi-thread synthesis engine, being 266MHz enough.

Table 3 contents a summary of the results of the real-time performance test.

	Single thread mean time per sample	Multithread mean time per sample
HP iPAQ 2210	21µs	99µs
HP iPAQ 6300	51µs	228µs
Dell Axim X50	17µs	83µs
Acer N35	30µs	102µs
Toshiba PXA263	29µs	86µs

Table 3: Results of the real-time performance test in different PDAs.

3.3.3. Database loading method

When using the MBROLA speech engine with the text-to-speech system, a sound unit database is needed to generate the voice. There are several methods to load this unit in memory, and performance can be different with each one.

The first method used is 'Preload', which loads the complete database in memory before the synthesis starts, so every unit is already in memory when necessary. The second one is 'OnDemand', where the required unit is loaded when it appears on the text. And the third one is "Cache_n": n units are loaded at a time. The test was carried out with a 3000 characters Basque text, the dynamic library and the HP iPAQ 2210 PDA, as well as the 1300Mhz and 64 Mb RAM Pc. For the "Cache_n" method, n=2 was taken. The results are shown in the table 4.

Database load method	PDA	PC
Preload	13µs	0,5µs
OnDemmand	19µs	0,6µs
Cache_2	22µs	1,1µs

Table 4: Mean sample generation time for the different database loading methods.

The database loading method that performs best is the preload, because when a unit is necessary it is already loaded, and it is not necessary to perform the loading action. The main drawback of the preload method is the long time needed to begin synthesis, necessary to preload the full database.

The memory usage also differs on both methods. While the memory usage in the preload and cache methods is about 1.5 Mb, with small differences when the text size changes, the Preload method allocates the full database in memory. The Basque synthesis unit database used in the experiments uses 4.18 Mb, so the memory usage for the preload method is more intensive.

4. Developed applications

Using the Text to Speech conversion capabilities, three different applications have been developed for the Pocket Pc 2003 operating system.

4.1. AhoTTS

The first developed application is a synthesis demonstrator. It mainly consists on a text box where words to be read aloud are typed, and the 'Synthesize' button, as in Figure 7. There are different configuration options, including different voices, and control of the pitch and the speed of the voice.

4.2. Task reader

Within the Pocket PC 2003 operating system there is specific software to control the pending and finished tasks. With little changes in the Windows CE registry, it is possible to combine the mentioned software with the AhoTTS voice capabilities, so tasks are read aloud.

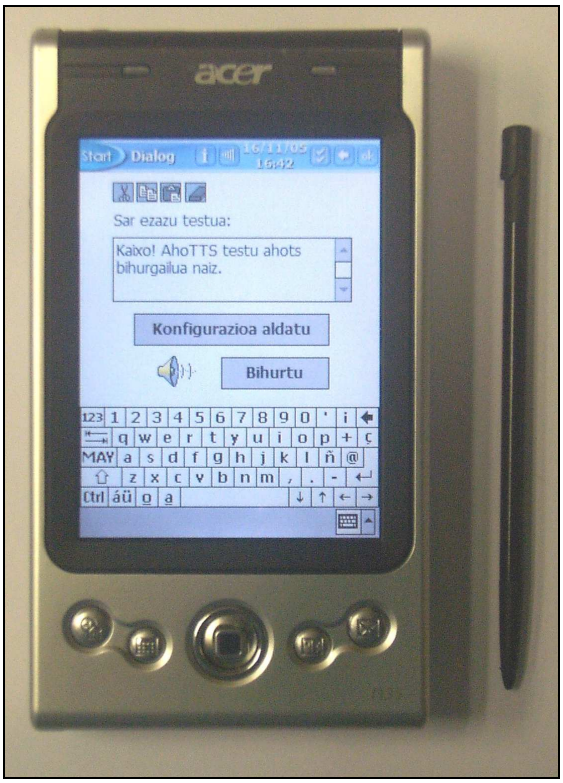

Image 7: The Acer N35 PDA running the AhoTTS demonstrator.

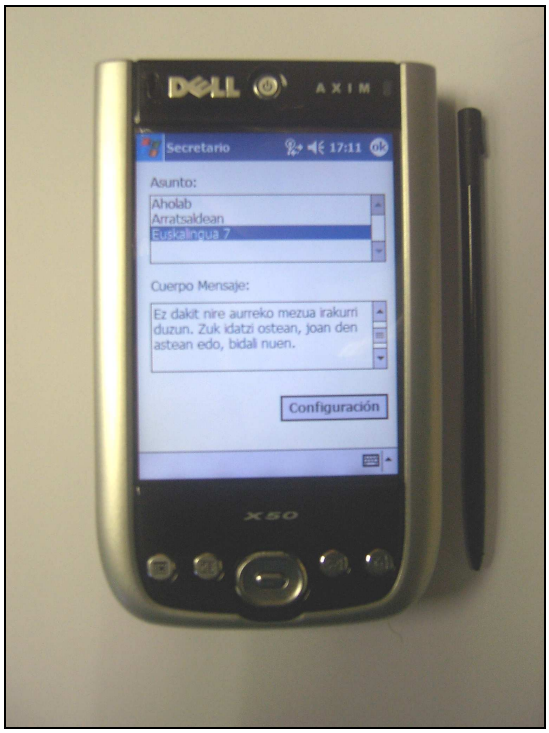

Image 8: *Secretario* e-mail reader, running in a DELL AXIM X50 PDA.

4.3. *Secretario*: e-mail reader

In the most recent PDA's, a wifi connection is usually included, which can provide Internet connection, and, consequently, web navigation or e-mail reading in the device itself. With this aim, a program that shows the received e-mails in the screen has been developed, with the particularity that tapping in one of the messages on the screen will cause the program to read it aloud.

4.4. Future work

With the developed DLL and API, several ideas appear for the future. The first one involves combining synthetic audio and video [7] in a single PDA.

The second project in process is the natural evolution of the e-mail reader. Nowadays the messages are played by a voice, but the user commands are tapped in the screen. The aim is to add voice recognition capabilities, so the e-mail program can use a completely oral interface, like the one developed for the PC – Windows XP architecture [8].

5. Acknowledgements

This work has been partially founded by the SAIOTEK program of the Basque Government, under the contract named S-PE04UN24 (PDATTS). The contribution of the Robotiker technological center has also been invaluable.

6. References

[1] Hernaez, I., Navas, E., Murugarren, J.L. and Etxebarria, B. "Description of the AhoTTS Conversion System for the Basque Language", *4th ISCA Tutorial and Research Workshop on Speech Synthesis*. 2001.

[2] http://bips.bi.ehu.es/SAMPA.html

[3] Charpentier, F. and Moulines, E., "Pitch-Synchronous waveform processing techniques for text-to-speech synthesis using diphones", *EUROSPEECH'89 Proceedings*

[4] Dutoit, T. and Leich, H., "MBR-PSOLA: Text-to-speech synthesis based on an MBE re-synthesis of the segments database". *Speech Communication, vol. 13, 1993, n. 3-4.*

[5] Microsoft Windows Mobile Homepage *http://www.microsoft.com/windowsmobile/*

[6] Dinkum Unabridged Library for VC++ *http://www.dinkumware.com/libdual_vc.html*

[7] Ortiz, A., Posada, J., Vivanco, K., Tejedor, M.G., Navas, E. and Hernáez, I., "Avatares Conversacionales 3D en Tiempo Real para su Integraci6n en Interfaces de Usuario y Entornos TV", *Proceedings II Jornada en Tecnologías del Habla*. Granada, Spain, 2002.

[8] Sainz, I., Navas, E., Sanchez, J., Luengo, I., and Hernáez, I., "Front-End for the Oral Control of Applications in Windows Environments" *Proceedings Eurocon 2005*, Belgrado. Serbia and Montenegro.