Multilingual TTS for Computer Telephony: 
The Aculab Approach

Alex Monaghan, Mahmoud Kassaei, Mark Luckin, Mariscela Amador-Hernandez, Andrew Lowry, 
Dan Faulkner, Fred Sannier

Aculab plc, Milton Keynes, UK
Alex.Monaghan@Aculab.com

Abstract
The requirements of the computer telephony (CT) industry place conflicting demands on text-to-speech (TTS) systems. Multilingual functionality and high quality output at telephone bandwidth requires detailed linguistic and acoustic analysis. At the same time, the need for robustness together with a high channel count and small memory footprint means that systems must be extremely efficient and databases must be kept small. We present a system which provides TTS for six languages, with 100 channels of highly natural output on a single DSP card.

1. Introduction
The Aculab multilingual TTS system has been developed over the past five years to provide unparalleled accuracy, quality and efficiency for CT applications. The current version supports six languages (UK and US English, French, German, Dutch, and Latin American Spanish), with multiple voices configurable for each language. Further languages and voices are under development. Telephone bandwidth versions (8kHz sampling rate) of all voices are available.

Aculab TTS is a concatenative synthesis system which uses a deliberately small database of natural speech to perform mixed-unit waveform concatenation. Any necessary modification of pitch and duration is performed by time-domain techniques.

Aculab's approach to multilingual TTS is based on three principles:

- The use of a multilingual system architecture, and multilingual modules wherever possible, so that language-specific elements are kept to a minimum;
- The application of detailed linguistic analysis and sophisticated symbolic processing, allowing us to produce accurate pronunciations and natural-sounding prosody;
- The careful design of speech databases, so that a small database contains a large number of units requiring minimal modification to produce high-quality speech.

The implementation of these three principles is discussed in the following three sections. A fourth section discusses evaluations of the performance of our TTS system, and a final section provides a summary and outlines our plans for future development.

Aculab TTS is specifically designed to run on Aculab's award-winning CT hardware, providing over 100 channels of telephone bandwidth speech on a single DSP card. The only restriction on the number of channels is the number of cards which can be accessed by the application. At the time of writing, Aculab TTS is available free of charge to users of Aculab's DSP hardware. For more information about these products, see Aculab's website at http://www.aculab.com.

Figure 1. Aculab TTS application architecture, and detail of TTS Server processing sequence.
includes a controller process which manages any number of servers. The client process sends requests to the controller or server, according to an Applications Programming Interface (API) which provides compatibility with other Aculab products and is compliant with various appropriate standards. Figure 1 illustrates a typical system architecture for an application using Aculab TTS. The API is documented at http://www.aculab.com/support_main/downloads_main.htm.

2.1. Commonality
All the languages which Aculab TTS supports are available on the same TTS server. They share a common system architecture, and they make use of the same modules in most cases. For example, there is a single program which performs letter-to-sound conversion for unknown words in all languages: this engine uses a separate rule file for each language, so adding a new language simply involves writing a new set of context-sensitive rewrite rules. The same is true of the postlexical rules, the syntactic analysis, and lower-level modules such as unit selection and modification. Some of these modules are discussed in more detail below.

The common system architecture is similar to classic rule-based TTS systems such as MITalk [1] and INFOVOX [2]: some important differences will be discussed below. Any particular language is synthesised by slotting the appropriate modules into this architecture, and perhaps adding or skipping some modules, as illustrated in Figure 1. The same common architecture also supplies most of the functions which are required by the modules: functions such as "get next word" or "compare syntactic categories" are built in, so that new modules can be written in an efficient and consistent manner. This approach also ensures that all data types are the same across languages, allowing us to produce language-independent modules and development tools.

2.2. Modularity
The Aculab TTS system is truly modular. Any set of modules may be executed in any order. This is probably more flexible than we need, but it means that modules may be included or omitted and that modules from different languages may be mixed for particular purposes (if, for example, we wish to synthesise a German text with French prosody).

Special modules may be added for particular languages, such as liaison processing for French, simply by inserting them at the appropriate point in the architecture. Similarly, modules may apply to one, some, or all languages as discussed in the section on prosody below.

2.3. Multilinguality
As far as possible, Aculab TTS processes all languages in the same way. We aim to exploit the linguistic commonalities between groups of languages, without compromising the naturalness or accuracy of our speech output.

2.4. Tools
In addition to the API for rapid application development, Aculab TTS provides three high-level tools for developers and content authors of CT applications: a lexicon manager, an email pre-processor, and a mark-up language.

The lexicon manager TTSLexMan allows users to build and maintain their own lexical resources, which can be used to override the standard lexica (to accommodate local or regional pronunciations) or to supplement the standard lexica with additional items such as customer names, products and brands.

The email pre-processor handles email headers intelligently, extracting the desired information and discarding other items which would not be suitable for synthesis. It also handles embedded messages, indented text, attached files, emoticons and many other phenomena peculiar to email.

The mark-up language is based on XML and W3C standards, and provides control of many aspects of the synthetic output: pronunciation, prosody, speech rate, etc. At the time of writing, this tool is not yet fully functional but it is scheduled for release in the third quarter of 2001.

3. Linguistic processing
In many ways, the linguistic processing in the Aculab TTS system is similar to classic rule-based TTS systems - for a review of such systems, see [3]. However, there are several features of Aculab TTS which are original and significantly different from other systems.

The basic sequence of modules shown in Figure 2 is not original, approximating as it does the process of human reading. The text normalisation, lexical look-up and letter-to-sound conversion are unremarkable, as are the postlexical rules and syllabification algorithms. The most original modules are the syntactic tagger and the prosody generator, so we will concentrate on those.

3.1. Tagging
Aculab TTS uses a bigram tagger to disambiguate parts of speech. The bigram probabilities are based on linguistic knowledge rather than corpus analysis, which gives them the following advantages:

- They can be easily edited and adapted to new domains or new languages;
- They can be adjusted to prefer particular sequences, so as to favour short phrases or increase the likelihood of nouns over verbs;
- Parts of speech can easily be added or deleted, to provide different granularities of analysis or to include semantic or prosodic classes.

We have argued elsewhere [4,5] that a full syntactic analysis is both impossible and unhelpful in a speech synthesis system. The Aculab TTS system performs syntactic analysis specifically to determine correct word pronunciations and to provide information for the prosodic rules: any analysis which does not serve these ends can be seen as superfluous.

In addition to syntactic tagging, we perform phrase-level "chunking" of the text: this is similar to the techniques used in [6], specifically to improve the prosody generation.

3.2. Prosody
There are five main stages in our prosody generation process:

- Assignment of major and minor prosodic domains;
- Assignment of emphasis within each domain;
- Calculation of segmental durations;
- Alignment of pitch targets with phonetic segments;
- Calculation of frequency values for pitch targets.
The last two stages are fully language-independent. Pitch target alignment follows the work cited in [7], which appears to generalise across all the languages in question. The calculation of frequency values is based on the original model in [8] and refined in [3]. In both cases, we believe that Aculab TTS is unusual in applying sophisticated phonological models to a commercial speech synthesis system. For the Germanic languages (currently English, Dutch or German in the current system) or a Romance language (French or Spanish, with Italian and Portuguese to be added in the near future), we will discuss both versions in some detail here: see [3] or [9] for an explanation of common prosodic terms. Figure 2 shows the multilingual design of the prosody generation process, with minimal language-specific elements.

3.2.1. Germanic Prosody

For the Germanic languages (currently English, Dutch and German), Aculab TTS assigns prosodic boundaries (boundary tones and pauses) and emphasis (stress and accent) by constructing a metrical tree [10]. The tree contains a subtree for each prosodic domain in the input text, and the terminal nodes in the tree are the stressable syllables in the domain. An initial tree is constructed based on syntactic information, lexical information, text punctuation and any mark-up specifications. This initial tree is then manipulated to produce a balanced, rhythmic assignment of emphases and boundaries. Translating the tree into a metrical grid [10] allows degrees of boundary and emphasis to be assigned, including a nuclear accent (or DTE) within each domain.

The assignment of segmental durations can then take account of the degree of emphasis on an item, and its proximity to a prosodic boundary, both of which have significant effects on the length of most segments. Until recently, the segmental durations for the Germanic languages in Aculab TTS have been based on statistics derived from corpus analysis, making this process the only truly stochastic element of our system. We have now moved away from a stochastic approach to segmental duration, and the current system uses a hybrid method combining intrinsic durations with a large number of modification factors. This change was necessary because of the very poor performance of the purely stochastic approach, and the segmental durations for Germanic languages are now much improved.

The application of metrical trees and grids to prosody generation in a commercial TTS system is probably unique to Aculab TTS, and allows us to drive both pitch and duration assignment from a sophisticated linguistic representation.

3.2.2. Romance Prosody

When the first Romance language (French) was added to Aculab TTS, we decided to implement a grid-only approach, i.e. to omit the construction of a metrical tree. This was partly due to the computational expense of tree construction and manipulation, and partly due to our belief that the computation of emphasis and boundaries for Romance languages is rather simpler than for the Germanic languages (at least at the level of accuracy which one can reasonably expect from a TTS system for unrestricted text). Romance languages tend not to indulge in deaccenting to the same extent as Germanic languages, and the location of the DTE is much less variable, resulting in a more regular assignment of prosodic emphases and boundaries.

We have based our assignment of emphases and boundaries in French on the work of Di Cristo and Hirst [11,12], and our Spanish algorithm is adapted from work by Garrido [13], Prieto and colleagues [14-16], and Karn [6]. In keeping with our multilingual approach, we have kept language dependent processes to a minimum (e.g. the application of the Bipolarisation Principle [12] at the word level in French, or the resolution of stress clashes which are preserved because of the absence of rhythmic deaccenting in Spanish [14]).

For Romance languages, Aculab TTS constructs a metrical grid based on syntactic, lexical and metatextual (punctuation and mark-up) information. The grid is then balanced according to language-independent rules, and the DTE is determined for each domain. Language-specific rhythm rules are then implemented. The resultant degrees of emphasis and boundary strength form the input to the duration rules and the language-independent pitch target alignment and scaling functions.

Segmental duration in Romance languages is calculated entirely by rule, with no stochastic component whatsoever. The grid provides a level of emphasis for each syllable, which translates into a target duration at the syllable level. A language-specific table of segmental durations provides an intrinsic duration for any syllable, and the mismatch between the intrinsic duration and the target duration results in modifications to the actual duration of individual segments in the particular syllable. The rules which govern these modifications are currently quite simple, and will be refined as more Romance languages are handled by the system. A single set of rules currently produces highly acceptable segmental durations for both French and Spanish, and we expect that this set of rules can be generalised to other Romance languages.

Again, Aculab’s application of current linguistic approaches to the generation of prosody is unique among commercial TTS systems and allows us to synthesise a wide range of prosodic variants in a very efficient manner. We expect to apply our grid-only approach to the Germanic languages in the near future.
4. Database design

The importance of database design is generally acknowledged by developers of concatenative TTS systems. In contrast to recent commercial TTS systems which have collected vast databases of relatively random speech, Aculab TTS uses small but very carefully designed databases. The reasons for this are threefold:

- We believe that a modest-sized database can capture all the necessary units for fully natural synthesis, if it is properly designed;
- We also believe that the speech quality from a smaller database (recorded over two or three days in a constant environment) is much more consistent than the quality of a much larger, and therefore less controlled, database;
- The nature of CT applications, and in fact of most other applications, makes small databases preferable to large ones.

Current Aculab TTS voices are recorded in a very controlled manner, and are designed to maximise consistency of voice quality. We use full sentences, including all relevant phonetic and prosodic contexts. We also record the speaker at multiple reference pitches, so as to minimise pitch modification during resynthesis.

Labelling is checked manually according to strict criteria, to ensure minimum spectral discontinuity at concatenation points. All speech is segmented and labelled as diphones, but larger units (i.e. sequences of contiguous diphones) are favoured by the unit selection algorithm, so that the number of concatenation points is minimised. We label as much of the material as possible, so that we have multiple units to choose from in most cases.

The run-time speech database for a typical voice recorded and labelled in this way is 16 megabytes, with an additional 2 megabytes of label files. These figures are for uncompressed 16bit speech at a sampling rate of 8kHz.

5. Evaluation

Aculab TTS has been evaluated by customers across three continents. Field trial results for Aculab TTS version 2.0 show that customers often preferred Aculab TTS compared with other current high-quality commercial TTS systems. We have not conducted any formal comparisons between Aculab TTS and other systems, as we would be open to accusations of bias. Instead, we have conducted an evaluation of Aculab TTS and other systems, as we would be open to accusations of bias.

Three major advantages of Aculab TTS are the number of channels of speech available (100 on a single DSP card), the small memory footprint, and the cost (currently free if used with appropriate Aculab DSP hardware).

Future versions of Aculab TTS will include new languages (Italian, Brazilian Portuguese and others), new voices for existing languages, improved waveform modification and concatenation techniques, and compliance with emerging standards for speech.

6. Summary

Aculab TTS is a multilingual TTS system designed for CT applications. It currently provides high-quality concatenative TTS for six languages. The quality is comparable with the best commercial TTS systems.

Three major advantages of Aculab TTS are the number of channels of speech available (100 on a single DSP card), the small memory footprint, and the cost (currently free if used with appropriate Aculab DSP hardware).

Future versions of Aculab TTS will include new languages (Italian, Brazilian Portuguese and others), new voices for existing languages, improved waveform modification and concatenation techniques, and compliance with emerging standards for speech.

7. References