A Short Introduction to Text-to-Speech Synthesis

Thierry Dutoit

2011

Abstract

I try to give here a short but comprehensive introduction to state-of-the-art Text-To-Speech (TTS) synthesis by highlighting its Digital Signal Processing (DSP) and Natural Language Processing (NLP) components. As a matter of fact, since very few people associate a good knowledge of DSP with a comprehensive insight into NLP, synthesis mostly remains unclear, even for people working in either research area.

After a brief definition of a general TTS system and of its commercial applications, in Section 1, the paper is basically divided into two parts. Section 2.1 begins with a presentation of the many practical NLP problems which have to be solved by a TTS system. I then examine, in Section 2.2, how synthetic speech can be obtained by simply concatenating elementary speech units, and what choices have to be made for this operation to yield high quality. I finaly give a word on existing TTS solutions, with special emphasis on the computational and economical constraints which have to be kept in mind when designing TTS systems.

Introduction

A Text-To-Speech (TTS) synthesizer is a computer-based system that should be able to read *any* text aloud, whether it was directly introduced in the computer by an operator or scanned and submitted to an Optical Character Recognition (OCR) system. Let us try to be clear. There is a fundamental difference between the system we are about to discuss here and any other talking machine (as a cassette-player for example) in the sense that we are interested in the automatic production of new sentences. This definition still needs some refinements. Systems that simply concatenate isolated words or parts of sentences, denoted as *Voice Response Systems*, are only applicable when a limited vocabulary is required (typically a few one hundreds of words), and when the sentences to be pronounced respect a very restricted structure, as is the case for the announcement of arrivals in train stations for instance. In the context of TTS synthesis, it is impossible (and luckily useless) to record and store all the words of the language. It is thus more suitable to define Text-To-Speech as the *automatic production of speech, through a grapheme-to-phoneme transcription of the sentences to utter.*

At first sight, this task does not look too hard to perform. After all, is not the human being potentially able to correctly pronounce an unknown sentence, even from his childhood ? We all have, mainly unconsciously, a deep knowledge of the reading rules of our mother tongue. They were transmitted to us, in a simplified form, at primary school, and we improved them year after year. However, it would be a bold claim indeed to say that it is only a short step before the computer is likely to equal the human being in that respect. Despite the present state of our knowledge and techniques and the progress recently accomplished in the fields of Signal Processing and Artificial Intelligence, we would have to express some reservations. As a matter of fact, the reading process draws from the furthest depths, often unthought of, of the human intelligence.

1. Automatic Reading: what for?

Each and every synthesizer is the result of a particular and original imitation of the human reading capability, submitted to technological and imaginative constraints that are characteristic of the time of its creation. The concept of *high quality TTS synthesis* appeared in the mid eighties, as a result of important developments in speech synthesis and natural language processing techniques, mostly due to the emergence of new technologies (Digital Signal and Logical Inference Processors). It is now a must for the speech products family expansion.

Potential applications of High Quality TTS Systems are indeed numerous. Here are some examples :

Telecommunications services. TTS systems make it possible to access textual information over the telephone. Knowing that about 70 % of the telephone calls actually require very little interactivity, such a prospect is worth being considered. Texts might range from simple messages, such as local cultural events not to miss (cinemas, the atres,...), to huge databases which can hardly be read and stored as digitized speech. Queries to such information retrieval systems could be put through the user's voice (with the help of a speech recognizer), or through the telephone keyboard (with DTMF systems). One could even imagine that our (artificially) intelligent machines could speed up the query when needed, by providing lists of keywords, or even summaries. In this connection, AT&T has recently organized a series of consumer tests for some promising telephone services [Levinson *et al.* 93]. They include: Who's Calling (get the spoken name of your caller before being connected and hang up to avoid the call), Integrated Messaging (have your electronic mail or facsimiles being automatically read over the telephone), Telephone Relay Service (have a telephone conversation with speech or hearing impaired persons thanks to *ad hoc* text-to-voice and voice-to-text conversion), and Automated Caller Name and Address (a computerized version of the "reverse directory"). These applications have proved acceptable, and even popular, provided the intelligibility of the synthetic utterances was high enough. Naturalness was not a major issue in most cases.

Language education. High Quality TTS synthesis can be coupled with a Computer Aided Learning system, and provide a helpful tool to learn a new language. To our knowledge, this has not been done yet, given the relatively poor quality available with commercial systems, as opposed to the critical requirements of such tasks.

Aid to handicapped persons. Voice handicaps originate in mental or motor/sensation disorders. Machines can be an invaluable support in the latter case : with the help of an especially designed keyboard and a fast sentence assembling program, synthetic speech can be produced in a few seconds to remedy these impediments. Astro-physician Stephen Hawking gives all his lectures in this way. The aforementioned Telephone Relay Service is another example. Blind people also widely benefit from TTS systems, when coupled with Optical Recognition Systems (OCR), which give them access to written information.

The market for speech synthesis for blind users of personal computers will soon be invaded by mass-market synthesisers bundled with sound cards. DECtalk (TM) is already available with the latest SoundBlaster (TM) cards now, although not yet in a form useful for blind people.

Talking books and toys. The toy market has already been touched by speech synthesis. Many speaking toys have appeared, under the impulse of the innovative 'Magic Spell' from Texas Instruments. The poor quality available inevitably restrains the educational ambition of such products. High Quality synthesis at affordable prices might well change this.

Vocal Monitoring. In some cases, oral information is more efficient than written messages. The appeal is stronger, while the attention may still focus on other visual sources of information. Hence the idea of incorporating speech synthesizers in measurement or control systems.

Multimedia, man-machine communication. In the long run, the development of high quality TTS systems is a necessary step (as is the enhancement of speech recognizers) towards more complete means of communication between men and computers. Multimedia is a first but promising move in this direction.

Fundamental and applied research. TTS synthesizers possess a very peculiar feature which makes them wonderful laboratory tools for linguists : they are completely under control, so that repeated experiences provide identical results (as is hardly the case with human beings). Consequently, they allow to investigate the efficiency of intonative and rhythmic models. A particular type of TTS systems, which are based on a description of the vocal tract through its resonant frequencies (its *formants*) and denoted as *formant synthesizers*, has also been extensively used by phoneticians to study speech in terms of acoustical rules. In this manner, for instance, articulatory constraints have been enlightened and formally described.

2. How does a machine read?

From now on, it should be clear that a reading machine would hardly adopt a processing scheme as the one naturally taken up by humans, whether it was for language analysis or for speech production itself. Vocal sounds are inherently governed by the partial differential equations of fluid mechanics, applied in a dynamic case since our lung pressure, glottis tension, and vocal and nasal tracts configuration evolve with time. These are controlled by our cortex, which takes advantage of the power of its parallel structure to extract the essence of the text read : its meaning. Even though, in the current state of the engineering art, building a Text-To-Speech synthesizer on such intricate models is almost scientifically conceivable (intensive research on articulatory synthesis, neural networks, and semantic analysis give evidence of it), it would result anyway in a machine with a very high degree of (possibly avoidable) complexity, which is not always compatible with economical criteria. After all, flies do not flap their wings !

Figure 1 introduces the functional diagram of a very general TTS synthesizer. As for human reading, it comprises a Natural Language Processing module (NLP), capable of producing a phonetic transcription of the text read, together with the desired intonation and rhythm (often termed as *prosody*), and a Digital Signal Processing module (DSP), which transforms the symbolic information it receives into speech. But the formalisms and algorithms applied often manage, thanks to a judicious use of mathematical and linguistic knowledge of developers, to short-circuit certain processing steps. This is occasionally achieved at the expense of some restrictions on the text to pronounce, or results in some reduction of the "emotional dynamics" of the synthetic voice (at least in comparison with human performances), but it generally allows to solve the problem in real time with limited memory requirements.

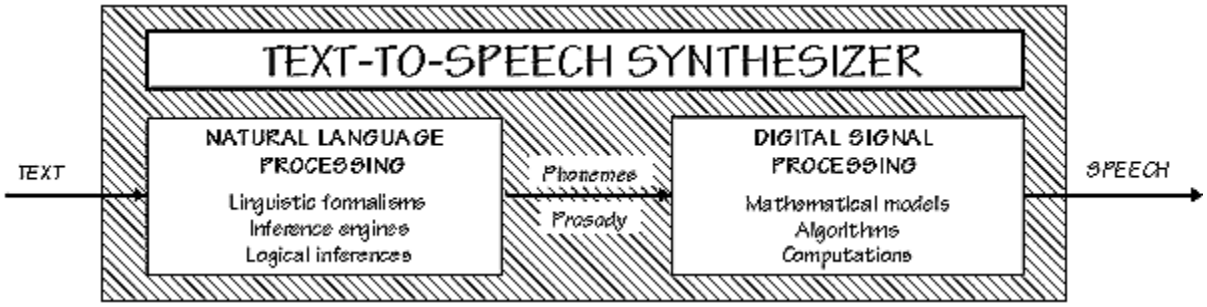

Figure 1. A simple but general functional diagram of a TTS system.

2.1. The NLP component

Figure 2 introduces the skeleton of a general NLP module for TTS purposes. One immediately notices that, in addition with the expected letter-to-sound and prosody generation blocks, it comprises a morpho-syntactic analyser, underlying the need for some syntactic processing in a high quality Text-To-Speech system. Indeed, being able to reduce a given sentence into something like the sequence of its parts-of-speech, and to further describe it in the form of a syntax tree, which unveils its internal structure, is required for at least two reasons :

1. Accurate phonetic transcription can only be achieved provided the part of speech category of some words is available, as well as if the dependency relationship between successive words is known.
2. Natural prosody heavily relies on syntax. It also obviously has a lot to do with semantics and pragmatics, but since very few data is currently available on the generative aspects of this dependence, TTS systems merely concentrate on syntax. Yet few of them are actually provided with full disambiguation and structuration capabilities.

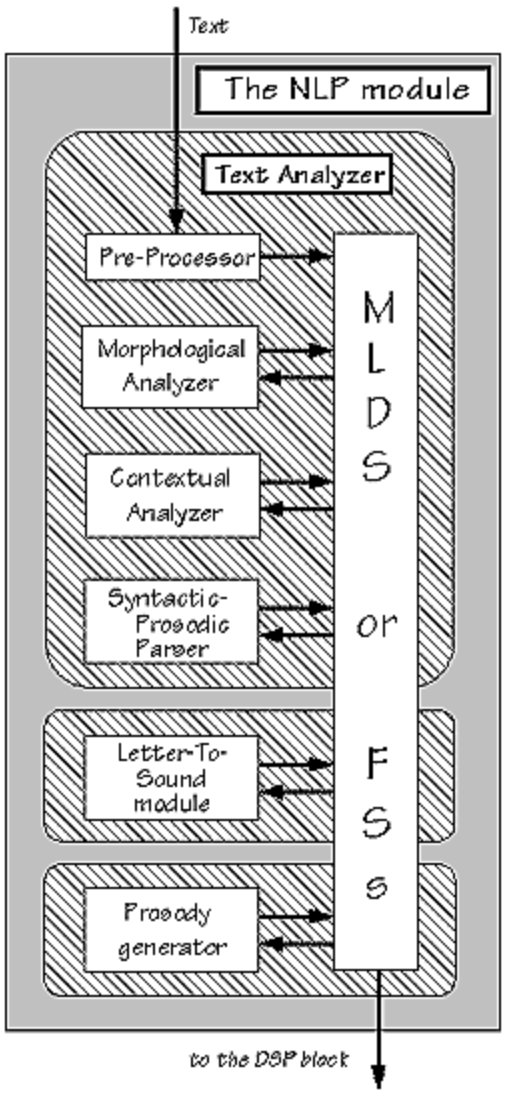

Fig 2. The NLP module of a general Text-To-Speech conversion system.

2.1.1. Text analysis

The text analysis block is itself composed of :

- A pre-processing module, which organizes the input sentences into manageable lists of words. It identifies numbers, abbreviations, acronyms and idiomatics and transforms them into full text when needed. An important problem is encountered as soon as the character level: that of punctuation ambiguity (including the critical case of sentence end detection). It can be solved, to some extent, with elementary regular grammars.
- A morphological analysis module, the task of which is to propose all possible part of speech categories for each word taken individually, on the basis of their spelling. Inflected, derived, and compound words are decomposed into their elementery graphemic units (their *morphs*) by simple regular grammars exploiting lexicons of stems and affixes

(see the CNET TTS conversion program for French [Larreur *et al.* 89], or the MITTALK system [Allen *et al.* 87]).

- The contextual analysis module considers words in their context, which allows it to reduce the list of their possible part of speech categories to a very restricted number of highly probable hypotheses, given the corresponding possible parts of speech of neighbouring words. This can be achieved either with *n-grams* [see Kupiec 92, Willemse & Gulikers 92, for instance], which describe local syntactic dependences in the form of probabilistic finite state automata (i.e. as a Markov model), to a lesser extent with *mutli-layer perceptrons* (i.e., neural networks) trained to uncover contextual rewrite rules, as in [Benello *et al.* 89], or with *local, non-stochastic grammars*provided by expert linguists or automatically inferred from a training data set with *classification and regression tree* (CART) techniques [Sproat *et al.* 92, Yarowsky 94].
- Finally, a syntactic-prosodic parser, which examines the remaining search space and finds the text structure (i.e. its organization into clause and phrase-like constituents) which more closely relates to its expected prosodic realization (see below).

2.1.2. Automatic phonetization

A poem of the Dutch high school teacher and linguist G.N. Trenite surveys this problem in an amusing way. It desperately ends with :

> *Finally, which rimes with "enough",*
> *Though, through, plough, cough, hough, or tough ?*
> *Hiccough has the sound of "cup",*
> *My advice is ... give it up !*

The Letter-To-Sound (LTS) module is responsible for the automatic determination of the phonetic transcription of the incoming text. It thus seems, at first sight, that its task is as simple as performing the equivalent of a dictionary look-up ! From a deeper examination, however, one quickly realizes that most words appear in genuine speech with several phonetic transcriptions, many of which are not even mentioned in pronunciation dictionaries. Namely :

1. Pronunciation dictionaries refer to word roots only. They do not explicitly account for morphological variations (i.e. plural, feminine, conjugations, especially for highly inflected languages, such as French), which therefore have to be dealt with by a specific component of phonology, called *morphophonology*.
2. Some words actually correspond to several entries in the dictionary, or more generally to several morphological analyses, generally with different pronunciations. This is typically the case of heterophonic homographs, i.e. words that are pronounced differently even though they have the same spelling, as for *'record'* (/rek**ù**d/ or/rIk**ù**d/), constitute by far the most tedious class of pronunciation ambiguities. Their correct pronunciation generally depends on their part-of-speech and most frequently contrasts verbs and non-verbs , as for '*contrast*' (verb/noun) or '*intimate*' (verb/adjective), although it may also be based on syntactic features, as for '*read*'(present/past)
3. Pronunciation dictionaries merely provide something that is closer to a *phonemic* transcription than from a*phonetic* one (i.e. they refer to phonemes rather than

to phones). As denoted by Withgott and Chen [1993] : "*while it is relatively straightforward to build computational models for morphophonological phenomena, such as producing the dictionary pronunciation of '*electricity*' given a baseform '*electric*', it is another matter to model how that pronunciation actually sounds"*. Consonants, for example, may reduce or delete in clusters, a phenomenon termed as *consonant cluster simplification*, as in *'softness'* [s�fnIs] in which [t] fuses in a single gesture with the following [n].

4. Words embedded into sentences are not pronounced as if they were isolated. Surprisingly enough, the difference does not only originate in variations at word boundaries (as with phonetic liaisons), but also on alternations based on the organization of the sentence into non-lexical units, that is whether into groups of words (as for phonetic lengthening) or into non-lexical parts thereof (many phonological processes, for instance, are sensitive to syllable structure).
5. Finally, not all words can be found in a phonetic dictionary : the pronunciation of new words and of many proper names has to be deduced from the one of already known words.

Clearly, points 1 and 2 heavily rely on a preliminary morphosyntactic (and possibly semantic) analysis of the sentences to read. To a lesser extent, it also happens to be the case for point 3 as well, since reduction processes are not only a matter of context-sensitive phonation, but they also rely on morphological structure and on word grouping, that is on morphosyntax. Point 4 puts a strong demand on sentence analysis, whether syntactic or metrical, and point 5 can be partially solved by addressing morphology and/or by finding graphemic analogies between words.

It is then possible to organize the task of the LTS module in many ways (Fig. 3), often roughly classified into *dictionary-based* and *rule-based* strategies, although many intermediate solutions exist.

Dictionary-based solutions consist of storing a maximum of phonological knowledge into a lexicon. In order to keep its size reasonably small, entries are generally restricted to morphemes, and the pronunciation of surface forms is accounted for by inflectional, derivational, and compounding morphophonemic rules which describe how the phonetic transcriptions of their morphemic constituents are modified when they are combined into words. Morphemes that cannot be found in the lexicon are transcribed by rule. After a first phonemic transcription of each word has been obtained, some phonetic post-processing is generally applied, so as to account for coarticulatory smoothing phenomena. This approach has been followed by the MITTALK system [Allen *et al*. 87] from its very first day. A dictionary of up to 12,000 morphemes covered about 95% of the input words. The AT&T Bell Laboratories TTS system follows the same guideline [Levinson *et al.* 93], with an augmented morpheme lexicon of 43,000 morphemes [Coker 85].

A rather different strategy is adopted in *rule-based* transcription systems, which transfer most of the phonological competence of dictionaries into a set of letter-to-sound (or *grapheme-to-phoneme*) rules. This time, only those words that are pronounced in such a particular way that they constitute a rule on their own are stored in an exceptions dictionary. Notice that, since many exceptions are found in the most frequent words, a reasonably small exceptions dictionary can

account for a large fraction of the words in a running text. In English, for instance, 2000 words typically suffice to cover 70% of the words in text [Hunnicut 80].

It has been argued in the early days of powerful dictionary-based methods that they were inherently capable of achieving higher accuracy than letter-to-sound rules [Coker *et al* 90], given the availability of very large phonetic dictionaries on computers. On the other hand, considerable efforts have recently been made towards designing sets of rules with a very wide coverage (starting from computerized dictionaries and adding rules and exceptions until all words are covered, as in the work of Daelemans & van den Bosch [1993] or that of Belrhali *et al* [1992]). Clearly, some trade-off is inescapable. Besides, the compromise is language-dependent, given the obvious differences in the reliability of letter-to-sound correspondences for different languages.

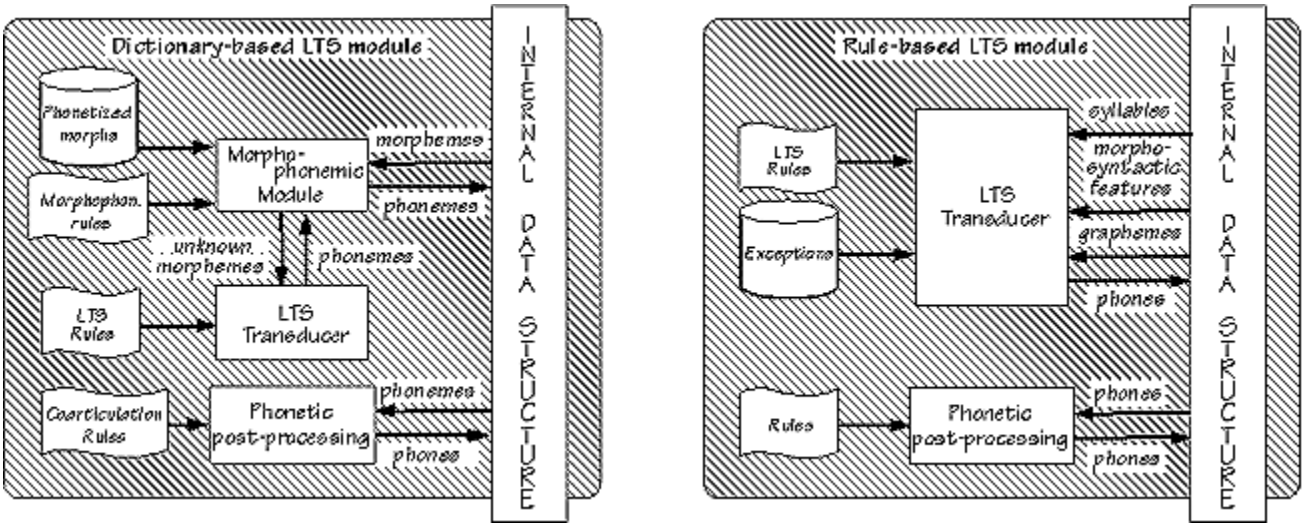

Fig. 3. Dictionary-based (left) versus rule-based (right) phonetization.

2.1.3. Prosody generation

The term *prosody* refers to certain properties of the speech signal which are related to audible changes in pitch, loudness, syllable length. Prosodic features have specific functions in speech communication (see Fig. 4). The most apparent effect of prosody is that of focus. For instance, there are certain pitch events which make a syllable stand out within the utterance, and indirectly the word or syntactic group it belongs to will be highlighted as an important or new component in the meaning of that utterance. The presence of a focus marking may have various effects, such as contrast, depending on the place where it occurs, or the semantic context of the utterance.

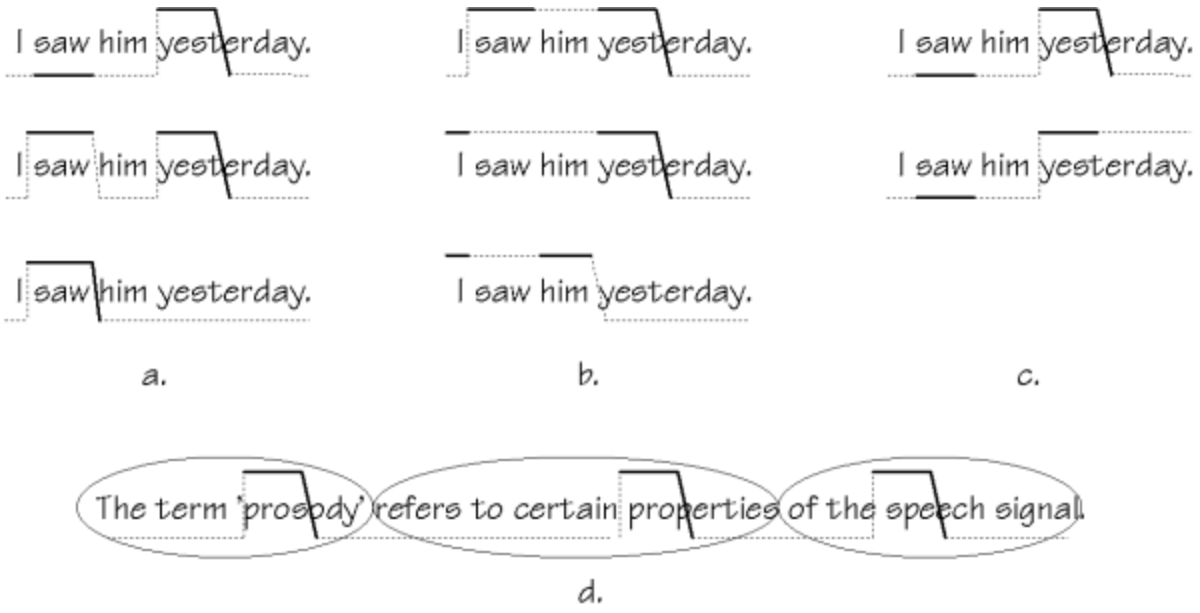

Fig. 4. Different kinds of information provided by intonation (lines indicate pitch movements; solid lines indicate stress).
a. Focus or given/new information;
b. Relationships between words (saw-yesterday; I-yesterday; I-him)
c. Finality (top) or continuation (bottom), as it appears on the last syllable;
d. Segmentation of the sentence into groups of syllables.

Although maybe less obvious, there are other, more systematic or general functions.

Prosodic features create a segmentation of the speech chain into groups of syllables, or, put the other way round, they give rise to the grouping of syllables and words into larger chunks. Moreover, there are prosodic features which indicate relationships between such groups, indicating that two or more groups of syllables are linked in some way. This grouping effect is hierarchical, although not necessarily identical to the syntactic structuring of the utterance.

So what ? Does this mean that TTS systems are doomed to a mere robot-like intonation until a brilliant computational linguist announces a working semantic-pragmatic analyzer for unrestricted text (i.e. not before long) ? There are various reasons to think not, provided one accepts an important restriction on the naturalness of the synthetic voice, i.e. that its intonation is kept '*acceptable neutral*' :

"Acceptable intonation must be plausible, but need not be the most appropriate intonation for a particular utterance : no assumption of understanding or generation by the machine need be made. Neutral intonation does not express unusual emphasis, contrastive stress or stylistic effects : it is the default intonation which might be used for an utterance out of context. (...) This approach removes the necessity for reference to context or world knowledge while retaining ambitious linguistic goals." [Monaghan 89]

The key idea is that the "correct" syntactic structure, the one that precisely requires some semantic and pragmatic insight, is not essential for producing such a prosody [see also O'Shaughnessy 90].

With these considerations in mind, it is not surprising that commercially developed TTS system have emphasized coverage rather than linguistic sophistication, by concentrating their efforts on text analysis strategies aimed to segment the *surface structure* of incoming sentences, as opposed to their syntactically, semantically, and pragmatically related *deep structure*. The resulting syntactic-prosodic descriptions organize sentences in terms of prosodic groups strongly related to phrases (and therefore also termed as *minor* or *intermediate phrases*), but with a very limited amount of embedding, typically a single level for these minor phrases as parts of higher-order prosodic phrases (also termed as *major*or *intonational phrases*, which can be seen as a prosodic-syntactic equivalent for clauses) and a second one for these major phrases as parts of sentences, to the extent that the related major phrase boundaries can be safely obtained from relatively simple text analysis methods. In other words, they focus on obtaining an acceptable segmentation and translate it into the continuation or finality marks of Fig. 4.c, but ignore the relationships or contrastive meaning of Fig. 4.a and b.

Liberman and Church [1992], for instance, have recently reported on such a very crude algorithm, termed as the *chinks 'n chunks* algorithm, in which prosodic phrases (which they call *f-groups*) are accounted for by the simple regular rule :

a (minor) prosodic phrase = a sequence of chinks followed by a sequence of chunks

in which *chinks* and *chunks* belong to sets of words which basically correspond to function and content words, respectively, with the difference that objective pronouns (like '*him*' or '*them*') are seen as chunks and that tensed verb forms (such as '*produced*') are considered as chinks. They show that this approach produces efficient grouping in most cases, slightly better actually than the simpler decomposition into sequences of function and content words, as shown in the example below:

function words / content words	**chinks / chunks**
I asked	***I asked them***
them if they were going home	***if they were going home***
to Idaho	***to Idaho***
and they said yes	***and they said yes***
and anticipated	***and anticipated one more stop***
one more stop	***before getting home*** **(6.7)**
before getting home **(6.6)**	

Other, more sophisticated approaches include syntax-based expert systems as in the work of [Traber 93] or [Bachenko & Fitzpatrick 90], and automatic, corpus-based methods as with the *classification and regression tree* (CART) techniques of Hirschberg [1991].

Once the syntactic-prosodic structure of a sentence has been derived, it is used to obtain the precise duration of each phoneme (and of silences), as well as the intonation to apply on them. This last step, however, is not straightforward either. It requires to formalize a lot of phonetic or phonological knowledge, either obtained from experts or automatically acquired from data with statistical methods. More information on this can be found in [Dutoit 96].

2.2. The DSP component

Intuitively, the operations involved in the DSP module are the computer analogue of dynamically controlling the articulatory muscles and the vibratory frequency of the vocal folds so that the output signal matches the input requirements. In order to do it properly, the DSP module should obviously, in some way, take articulatory constraints into account, since it has been known for a long time that phonetic transitions are more important than stable states for the understanding of speech [Libermann 59]. This, in turn, can be basically achieved in two ways :

- Explicitly, in the form of a series of rules which formally describe the influence of phonemes on one another;
- Implicitly, by storing examples of phonetic transitions and co-articulations into a speech segment database, and using them just as they are, as ultimate acoustic units (i.e. in place of phonemes).

Two main classes of TTS systems have emerged from this alternative, which quickly turned into synthesis philosophies given the divergences they present in their means and objectives : *synthesis-by-rule* and *synthesis-by-concatenation*.

2.2.1. Rule-based synthesizers

Rule-based synthesizers are mostly in favour with phoneticians and phonologists, as they constitute a cognitive, generative approach of the phonation mechanism. The broad spreading of the Klatt synthesizer [Klatt 80], for instance, is principally due to its invaluable assistance in the study of the characteristics of natural speech, by analytic listening of rule-synthesized speech. What is more, the existence of relationships between articulatory parameters and the inputs of the Klatt model make it a practical tool for investigating physiological constraints [Stevens 90].

For historical and practical reasons (mainly the need for a physical interpretability of the model), rule synthesizers always appear in the form of *formant synthesizers*. These describe speech as the dynamic evolution of up to 60 parameters [Stevens 90], mostly related to formant and anti-formant frequencies and bandwidths together with glottal waveforms. Clearly, the large number of (coupled) parameters complicates the analysis stage and tends to produce analysis errors. What is more, formant frequencies and bandwidths are inherently difficult to estimate from speech data. The need for intensive trials and errors in order to cope with analysis errors, makes them time-consuming systems to develop (several years are commonplace). Yet, the synthesis

quality achieved up to now reveals typical buzzyness problems, which originate from the rules themselves : introducing a high degree of naturalness is theoretically possible, but the rules to do so are still to be discovered.

Rule-based synthesizers remain, however, a potentially powerful approach to speech synthesis. They allow, for instance, to study speaker-dependent voice features so that switching from one synthetic voice into another can be achieved with the help of specialized rules in the rule database. Following the same idea, synthesis-by-rule seems to be a natural way of handling the articulatory aspects of changes in speaking styles (as opposed to their prosodic counterpart, which can be accounted for by concatenation-based synthesizers as well). No wonder then that it has been widely integrated into TTS systems (MITTALK [Allen *et al.* 87] and the JSRU synthesizer [Holmes *et al.* 64] for English, the multilingual INFOVOX system [Carlson *et al.* 82], and the I.N.R.S system [O'Shaughnessy 84] for French).

2.2.2. Concatenative synthesizers

As opposed to rule-based ones, *concatenative synthesizers* possess a very limited knowledge of the data they handle : most of it is embedded in the segments to be chained up. This clearly appears in figure 6, where all the operations that could indifferently be used in the context of a music synthesizer (i.e. without any explicit reference to the inner nature of the sounds to be processed) have been grouped into a *sound processing* block, as opposed to the upper *speech processing* block whose design requires at least some understanding of phonetics.

Database preparation

A series of preliminary stages have to be fulfilled before the synthesizer can produce its first utterance. At first, segments are chosen so as to minimize future concatenation problems. A combination of diphones (i.e. units that begin in the middle of the stable state of a phone and end in the middle of the following one), half-syllables, and triphones (which differ from diphones in that they include a complete central phone) are often chosen as speech units, since they involve most of the transitions and co-articulations while requiring an affordable amount of memory. When a complete list of segments has emerged, a corresponding list of words is carefully completed, in such a way that each segment appears at least once (twice is better, for security). Unfavourable positions, like inside stressed syllables or in strongly reduced (i.e. over-co-articulated) contexts, are excluded. A corpus is then digitally recorded and stored, and the elected segments are spotted, either manually with the help of signal visualization tools, or automatically thanks to segmentation algorithms, the decisions of which are checked and corrected interactively. A segment database finally centralizes the results, in the form of the segment names, waveforms, durations, and internal sub-splittings. In the case of diphones, for example, the position of the border between phones should be stored, so as to be able to modify the duration of one half-phone without affecting the length of the other one.

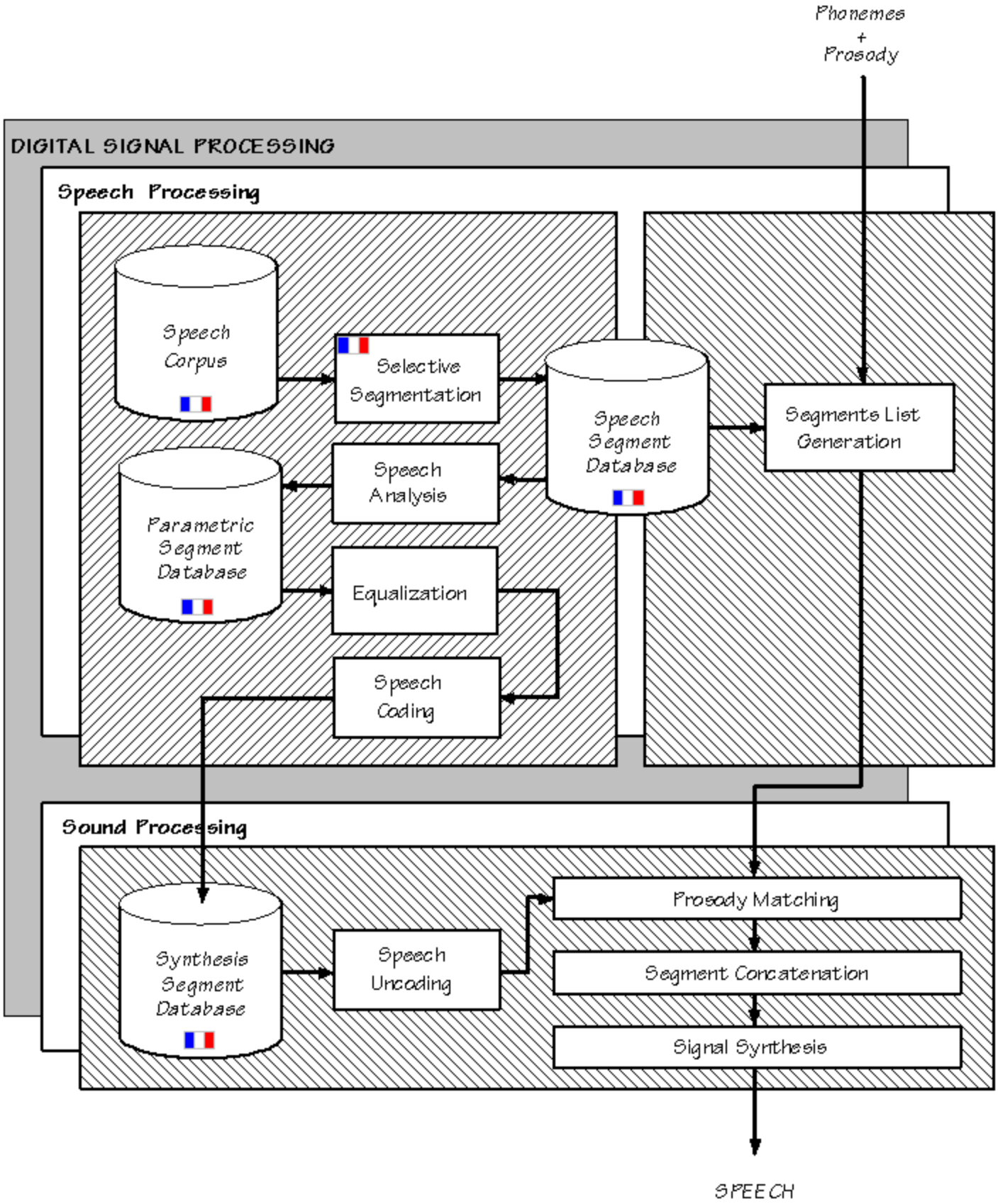
Phonemes
+
Prosody
DIGITAL SIGNAL PROCESSING
Speech Processing
Speech
Corpus
Selective
Segmentation
Speech
Segment
Database
Segments List
Generation
Speech
Analysis
Parametric
Segment
Database
Equalization
Speech
Coding
Sound Processing
Synthesis
Segment
Database
Speech
Uncoding
Prosody Matching
Segment Concatenation
Signal Synthesis
SPEECH

Figure 5. A general concatenation-based synthesizer. The upper left hatched block corresponds to the development of the synthesizer (i.e. it is processed once for all). Other blocks correspond to run-time operations. Language-dependent operations and data are indicated by a flag.

Segments are then often given a parametric form, in the form of a temporal sequence of vectors of parameters collected at the output of a *speech analyzer* and stored in a *parametric* segment database. The advantage of using a speech model originates in the fact that :

- Well chosen speech models allow data size reduction, an advantage which is hardly negligible in the context of concatenation-based synthesis given the amount of data to be stored. Consequently, the analyzer is often followed by a parametric *speech coder*.
- A number of models explicitly separate the contributions of respectively the source and the vocal tract, an operation which remains helpful for the pre-synthesis operations : prosody matching and segments concatenation.

Indeed, the actual task of the synthesizer is to produce, in real-time, an adequate sequence of concatenated segments, extracted from its parametric segment database and the prosody of which has been adjusted from their stored value, i.e. the intonation and the duration they appeared with in the original speech corpus, to the one imposed by the language processing module. Consequently, the respective parts played by the prosody matching and segments concatenation modules are considerably alleviated when input segments are presented in a form that allows easy modification of their pitch, duration, and spectral envelope, as is hardly the case with crude waveform samples.

Since segments to be chained up have generally been extracted from different words, that is in different phonetic contexts, they often present amplitude and timbre mismatches. Even in the case of stationary vocalic sounds, for instance, a rough sequencing of parameters typically leads to audible discontinuities. These can be coped with during the constitution of the synthesis segments database, thanks to an *equalization* in which related endings of segments are imposed similar amplitude spectra, the difference being distributed on their neighbourhood. In practice, however, this operation, is restricted to amplitude parameters : the equalization stage smoothly modifies the energy levels at the beginning and at the end of segments, in such a way as to eliminate amplitude mismatches (by setting the energy of all the phones of a given phoneme to their average value). In contrast, timbre conflicts are better tackled at run-time, by *smoothing* individual couples of segments when necessary rather than equalizing them once for all, so that some of the phonetic variability naturally introduced by co-articulation is still maintained. In practice, amplitude equalization can be performed either before or after speech analysis (i.e. on crude samples or on speech parameters).

Once the parametric segment database has been completed, synthesis itself can begin.

Speech synthesis

A sequence of segments is first deduced from the phonemic input of the synthesizer, in a block termed as *segment list generation* in Fig. 5, which interfaces the NLP and DSP modules. Once

prosodic events have been correctly assigned to individual segments, the *prosody matching* module queries the synthesis segment database for the actual parameters, adequately uncoded, of the elementary sounds to be used, and adapts them one by one to the required prosody. The segment concatenation block is then in charge of dynamically matching segments to one another, by smoothing discontinuities. Here again, an adequate modelization of speech is highly profitable, provided simple interpolation schemes performed on its parameters approximately correspond to smooth acoustical transitions between sounds. The resulting stream of parameters is finally presented at the input of a synthesis block, the exact counterpart of the analysis one. Its task is to produce speech.

Segmental quality

The efficiency of concatenative synthesizers to produce high quality speech is mainly subordinated to :

1. The type of segments chosen.

   Segments should obviously exhibit some basic properties :

   - They should allow to account for as many co-articulatory effects as possible.
   - Given the restricted smoothing capabilities of the concatenation block, they should be easily connectable.
   - Their number and length should be kept as small as possible.
   - On the other hand, longer units decrease the density of concatenation points, therefore providing better speech quality. Similarly, an obvious way of accounting for articulatory phenomena is to provide many variants for each phoneme. This is clearly in contradiction with the limited memory constraint. Some trade-off is necessary. Diphones are often chosen. They are not too numerous (about 1200 for French, including lots of phoneme sequences that are only encountered at word boundaries, for 3 minutes of speech, i.e. approximately 5 Mbytes of 16 bits samples at 16 kHz) and they do incorporate most phonetic transitions. No wonder then that they have been extensively used. They imply, however, a high density of concatenation points (one per phoneme), which reinforces the importance of an efficient concatenation algorithm. Besides, they can only partially account for the many co-articulatory effects of a spoken language, since these often affect a whole phone rather than just its right or left halves independently. Such effects are especially patent when somewhat transient phones, such as liquids and (worst of all) semi-vowels, are to be connected to each other. Hence the use of some larger units as well, such as triphones.
2. The model of speech signal, to which the analysis and synthesis algorithms refer.

The models used in the context of concatenative synthesis can be roughly classified into two groups, depending on their relationship with the actual phonation process. *Production models* provide mathematical substitutes for the part respectively played by vocal folds, nasal and vocal tracts, and by the lips radiation. Their most representative members are Linear Prediction Coding (LPC) synthesizers [Markel & Gray 76], and the formant synthesizers we

mentioned in section 2.2.1. On the contrary,*phenomenological models* intentionally discard any reference to the human production mechanism. Among these pure digital signal processing tools, spectral and time-domain approaches are increasingly encountered in TTS systems. Two leading such models exist : the hybrid Harmonic/Stochastic (H/S) model of [Abrantes *et al.* 91] and the Time-Domain Pitch-Synchronous-OveraLap-Add (TD-PSOLA) one [Moulines & Charpentier 90]. The latter is a time-domain algorithm : it virtually uses no speech explicit speech model. It exhibits very interesting practical features : a very high speech quality (the best currently available) combined with a very low computational cost (7 operations per sample on the average). The hybrid Harmonic/stochastic model is intrinsically more powerful than the TD-PSOLA one, but it is also about ten times more computationally intensive. PSOLA synthesizers are now widely used in the speech synthesis community. The recently developed MBROLA algorithm [Dutoit 93,96] even provides a time-domain algorithm which exhibits the very efficient smoothing capabilities of the H/S model (for the spectral envelope mismatches that cannot be avoided at concatenation points) as well as its very high data compression ratios (up to 10 with almost no additional computational cost) while keeping the computational complexity of PSOLA.

3. Conclusion

Let us bow to the facts : there is still a long way to HAL, the brilliant talking computer of *'2001, a space odyssey'*. A number of advances in the area of NLP or DSP, however, have recently boosted up the quality and naturalness of available voices, and this is likely to continue. Important issues need now be further addressed in that purpose. Among others :

- How to best account for coarticulatory phenomena ? In the context of concatenation-based synthesis, this question mostly reduces to : how to derive *optimized sets of segments* from speech data ?
- How to best formalize the relationship between syntax, semantics, pragmatics and prosody, and how to derive natural sounding intonation and duration from abstract prosodic patterns ?
- A fundamental feature of speech has seldom been taken into consideration by TTS systems : its *variability*. Prosodic patterns, for instance, are submitted to a particular kind of variability which cannot be confused with randomness in that variations maintain some hidden coherency with each other.
- How to account for speaker and speaking style effects ?

Readers willing to have a deeper understanding of the problems mentioned in this paper could advantageously report to the forthcoming [Dutoit 96], which analyses DSP and NLP solutions with much more details. A number of internet sites can also be consulted, some of which propose demo programs and/or speech files. See for example the speech synthesis virtual museum at URL address :

http://www.cs.bham.ac.uk/~jpi/synth/museum.html

http://tcts.fpms.ac.be/synthesis