

Chapter 2

Speech Synthesis

This chapter gives an introduction to speech synthesis. A general structure of TTS systems is introduced and the four main steps for producing a synthetic speech signal are explained. The main focus is put upon different methods for the speech signal generation, namely: parametric methods, concatenative speech synthesis, model-based synthesis approaches and hybrid models. Moreover, distortions that are specific for these systems are discussed. Finally, the open-source MaryTTS system is introduced.

2.1 Setup of a Speech Synthesizer

Different approaches towards synthetic speech have been developed over the years. So called *canned speech* systems use prerecorded phrases and play them back without, or only with very little changes. Such systems are therefore only employed in limited-domain systems, e.g., train station announcements. A more sophisticated approach is taken by Concept-to-Speech (CTS) systems. The main idea of CTS is the generation of waveforms directly from a linguistic description, i.e., without any specified input text. Although the idea of bypassing the difficulties of natural language processing seems promising, CTS systems have not managed to play an important role in speech synthesis.

Considering the limitations of *canned speech* and the shortcomings of CTS, this book exclusively concentrates on Text-to-Speech (TTS) synthesis. Even though there is a multitude of different approaches to TTS, there are similar steps every system has to go through in order to produce a synthetic speech signal. A general structure for a TTS system can be seen in Fig. 2.1. Such systems usually consist of four main steps: natural language processing, prosody generation, concatenation, and speech generation. These four steps are explained in the following sections.

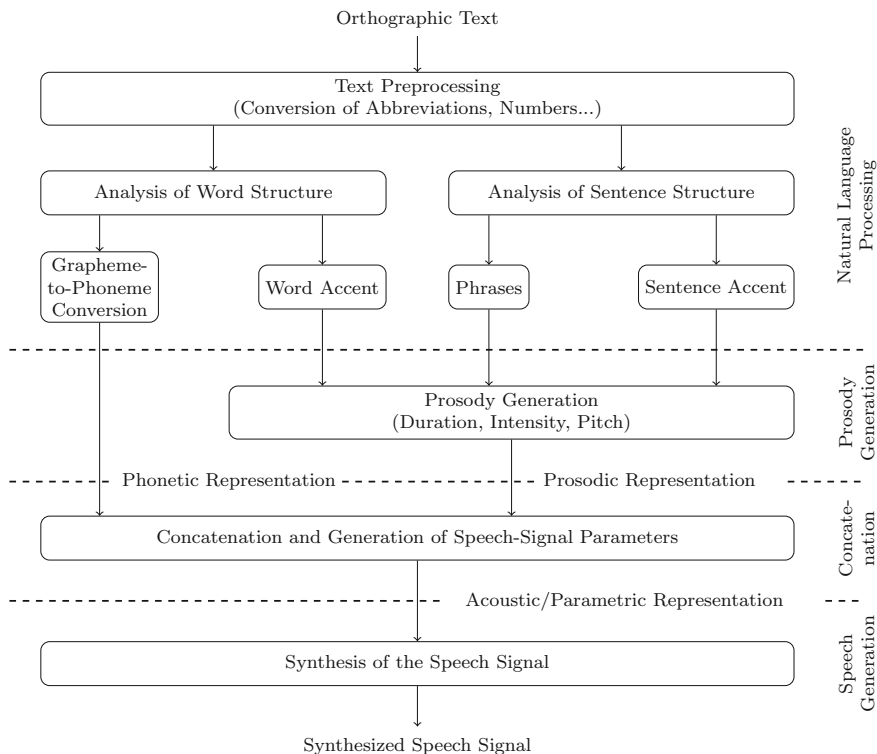


Fig. 2.1 General structure of TTS systems based on [1]

2.1.1 Natural Language Processing (NLP)

The first module of every TTS system is a text preprocessing unit. It analyzes the orthographic text and identifies special cases, e.g., abbreviations, numbers, foreign language terms, proper names etc. Those cases require special treatment, e.g., numbers have to be transformed into written text.

The preprocessing step is followed by an analysis of word and sentence structure. Therefore, on the one hand a morphologic analysis identifies word stems, prefixes, and suffixes which are important for the word stress. On the other hand, the sentence level structure is evaluated. This reveals sentence accents and information on phrases (i.e., groups of words). Word and sentence accents as well as identified phrases are needed in the following for the prosody generation.

Moreover, the morphologic output is used by the Grapheme-to-Phoneme (G2P) unit to transform the graphemes into their corresponding phonemes and thus create a phonetic representation of the orthographic text in a computer-readable format, e.g., using the (SAMPA) [1, 2].

2.1.2 Prosody Generation

The prosody generation unit uses information on word and sentence accents along with information on phrases to create the corresponding prosody of the orthographic text, i.e., duration, intensity and pitch.

To achieve this, procedures like the Fujisaki model [3, 4] can be applied. This model describes a pitch contour as a superposition of *phrase commands* and *accent commands* and an underlying *base frequency*. A detailed view on the Fujisaki model and features that can be derived from it is given in Sect. 6.2.2.1.

2.1.3 Concatenation and Generation of Speech-Signal Parameters

Based on the phonetic representation of the orthographic text generated by the G2P unit and the prosodic information, the concatenation unit creates a continuous sequence of signal parameters and/or articulation gestures.

Those first three units (NLP, prosody generation, and concatenation) solely depend on the given orthographic input text and are thus completely independent of the speaker of the speech signal to be synthesized.

2.1.4 Speech Signal Generation

This section introduces different methods for the speech signal generation within a TTS system based on the continuous sequence of acoustic units and/or signal parameters.

2.1.4.1 Parametric Speech Synthesis

In contrast to all other synthesis techniques that will be discussed in the following sections, parametric synthesizers do not use any prerecorded speech data, they generate the speech signal solely based on their input parameters. The basic principle behind this approach is the source-filter model of the human voice production system as shown in Fig. 2.2. According to this model, a speech signal can be generated from a voiced or an unvoiced excitation signal (or a combination of both) and parameters that describe the filter characteristics of the vocal tract, e.g., through parameters that reflect the shape of the pharynx and the mouth, the position of the tongue, and the rounding of the lips etc. Thus, these parameters specify a simple model of the vocal tract geometry. This model can then be used for an acoustic simulation of the human speech production. Such articulatory synthesizers [5] are still being developed,

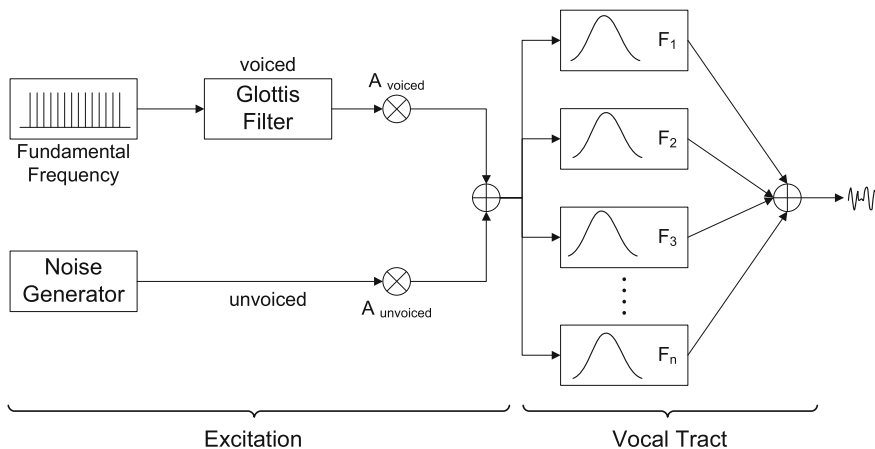


Fig. 2.2 Source-filter model of the human voice production

however, due to their inferior quality and their high computing time, they are mainly used for research.

An easier way to implement a parametric speech synthesizer is the realization of the vocal tract by formant filters.¹ These so called (FO) synthesizers are still being used, especially in systems with strong memory restrictions.

Formant Synthesizer

First attempts towards a synthesis based on formant filters have already been made in the early 1960s [6] and stayed popular long after that. The development of FO synthesis peaked in the late 80s/early 90s just before TTS research turned towards the concatenation of speech units. During this time, the Klatt synthesizer [7] was one of the state-of-the-art systems. Klatt's FO synthesis [8] (and FO synthesis in general) closely follows the source-filter model in which an impulse train is used to model voiced parts of a signal while white noise is the basis for the excitation of unvoiced sounds. Combining both excitation types adds breathiness to the generated speech signal and is used for the creation of fricatives. The filter itself can be realized through 2nd order IRR filters which represent the resonances of the vocal tract. In Klatt's FO synthesizer, a cascading of several sections (cascading model) is used to generate voiced sounds while adding several sections together (parallel model) is used to synthesize fricatives and stops.

While FO synthesizers are able to create “clean” sounding, intelligible speech, the achieved quality is far from that of natural speech. Reasons for this are the too simplistic models for both the excitation signal as well as the vocal tract [9]. Typical artifacts of FO synthesis are “metallic” voices that sound very artificial. Therefore, FO synthesizers cover the lower end of TTS quality of the synthesizers discussed in this chapter. Nonetheless, due to their parametric nature, FO synthesizers are easily

¹Formants are peaks in the envelope of speech sounds and thus define the characteristics of a sound.

customizable, e.g., modifications of speech rate can be executed by simple parameter changes. This is an important feature, especially for visually impaired people who prefer acoustic cues rather than written text. By increasing the speech rate they are able to comprehend even faster [10]. Studies have shown that blind people are even able to comprehend ultra-fast synthetic speech at speaking rates above 17 syllables per second while a normal speech rate features only 3–5 syllables per second [11].

2.1.4.2 Concatenative Speech Synthesis

The development of concatenative speech synthesizers brought a great leap forward for the quality of TTS. By playing back prerecorded speech samples, the quality of synthesized speech was now theoretically able to equal that of natural speech. In reality, however, the quality of concatenative systems varies greatly: If the corpus contains units that are close to the text to be synthesized, the quality can be almost human-like. If that is not the case, severe distortions can occur. Thus, the inventory design is a crucial task. In most cases tailor-made speech corpora are recorded to fit the context of the systems.

The choice of unit size is one of the most important decisions for a TTS system. Finding the right balance between a preferably small footprint and a quality that is sufficient for the use case is a challenging task. At first sight it may seem that building a speech corpus based on the phonemes of a language, e.g., 42 phonemes for English, would cover every necessary sound. In practice, however, this would lead to major distortions at the transitions between the units. This is due to the coarticulation effect, i.e., the influence of a phone on its proceeding phone. Therefore, the smallest unit used in speech synthesis is the diphone, i.e., a unit that starts at the center of a phone and ends at the center of the following phone and thus covers the coarticulation. For the English language this leads to an approximate number of units of about 1500 diphones. Longer units, e.g., triphones, demisyllables, syllables etc., yield superior quality with the drawback of a much larger database. Building a TTS using, for example, a word inventory, would cause between 100 K and 1.5 M units [8].

But even with longer units synthesized speech still contains distortions that make it sound artificial. These distortions are due to two types of discontinuities [8]:

- *spectral discontinuities* occur when the formants of two aligned units do not match;
- *prosodic discontinuities* emerge from unfitting pitch curves at the concatenation point.

There are two different approaches to deal with these discontinuities:

- A technique called *Pitch Synchronous Overlap and Add (PSOLA)* manipulates the units in the time or spectral domain to correct these discontinuities.
- *Unit Selection (US) synthesis* uses an inventory that contains multiple instances of every unit with different pitch and formant curves and chooses the most appropriate one during concatenation.

Both approaches are discussed in the following sections.

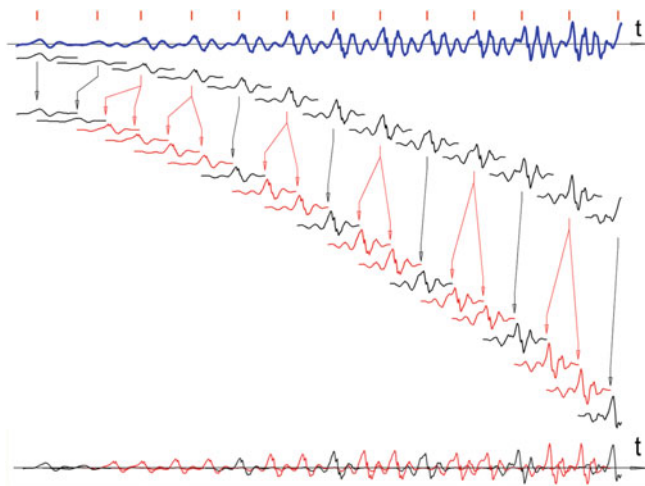


Fig. 2.3 Using PSOLA to increase the pitch of a signal by 50% [12]

Pitch Synchronous Overlap and Add (PSOLA)

In order to ensure a smooth transition, i.e., eliminating spectral and prosodic discontinuities, amplitude, duration and pitch of conjoined units have to be adjusted. The amplitude can easily be varied just by multiplying the waveform to the desired value. Applying PSOLA makes it possible to also change duration and pitch of each segment. In general, however, these modifications introduce distortions in each unit. To keep them as small as possible the most common PSOLA approach operates in the time domain (TD-PSOLA). This still causes degradations, but the gain of prosodically modified concatenated units usually is greater than the introduced distortions.

An example of how PSOLA works is shown in Fig. 2.3. Here the pitch gets increased by 50%. The upper part of the figure shows a diphone unit in blue with the red ticks marking its pitch cycles. A window function is applied which splits up the original diphone unit into smaller PSOLA units that cover two pitch cycles (waveforms in black). By doubling half of the PSOLA units (waveforms in red) while keeping the duration of the whole diphone unit constant the resulting diphone (waveform at the bottom of the figure) then features a pitch increased by 50%.

A similar technique is implemented in the MBROLA algorithm which has reached popularity due to its non-commercial availability. Since both approaches are very similar and are necessary for diphone concatenation they will both be referred to as (DI) systems.

Other, less common approaches apply PSOLA in the in the Frequency Domain (FD-PSOLA) or via Linear Prediction (LP-PSOLA). Given their high computational complexity, these approaches are rarely used.

Compared to the parametric approaches, PSOLA generates a decent quality which is moreover more independent from the context to be synthesized than the US

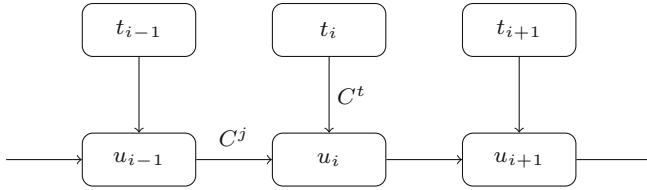


Fig. 2.4 Unit selection costs [15]

approach which will be described in the next section. Furthermore, PSOLA can operate on a reasonably small inventory. The downside of this technique is its sensitivity to accurate pitch marks. Moreover, changes in pitch are limited to ± 0.5 octaves and spectral discontinuities can only be corrected by the computationally more complex LP-PSOLA [12].

Unit Selection (US) Synthesizer

Even though PSOLA synthesis yields decent quality, the signal manipulations that come with this approach induce new distortions. In order to overcome the need for adjustments in speech units, Black et al. introduced the Unit Selection (US) approach for their synthesis platform CHATR [13, 14]. In contrast to PSOLA, a US synthesizer stores a huge database of speech units. Therefore, during the concatenation process a US system can choose from a large pool of candidate units. Each unit in the database has an attached feature vector that describes, e.g., phonetic and prosodic context (duration, power, and pitch) and acoustic join costs. This enables a synthesizer to pick units that guarantee smooth pitch and formant transitions. Hence, the main challenge for a US system is to select the most appropriate units.

The basis for the selection of units are the two cost functions [15, 16] shown in Fig. 2.4. Here, a unit from the database is denoted by u_i whereas a target unit is given by t_i . Thus the cost functions are defined as follows:

- *target costs* $C^t(u_i, t_i)$ are given as the difference between a database unit u_i and a target unit t_i which it should represent
- *join costs*² $C^j(u_{i-1}, u_i)$ estimate the quality of two joined units (u_{i-1} and u_i) and can be computed *offline*, before the actual synthesis takes place.

Thus, the process of unit selection means to find a balance between target and join costs without putting too much emphasis on one of them. An optimization of this problem leads to smooth transitions between all chosen units and thereby also to a minimum of discontinuities for pitch and formant curves of the respective speech database.

While a pure US system concatenates the units from the database without any signal manipulations, newer approaches shape both pitch and formant curves in

²Also known as *concatenation costs*.

order to ensure even smoother transitions. However, these manipulations occur far less often than in PSOLA synthesis.

The main advantage of the US approach is its perceptually high quality. However, to achieve high quality with a US system, the available units have to match the context to be synthesized. If this is not the case, glitches may occur and thus the quality will be severely degraded. Even a single glitch in an otherwise perfectly synthesized sentence will spoil the perceptual impression of a listener. Therefore, the quality of a US system strongly depends on the context and can vary drastically over time. Moreover, if there are no modifications to the units in the database, the synthesized speech is limited to the speaking style of the original recordings. More sophisticated approaches include a PSOLA algorithm to make slight changes to the units before concatenation in order to guarantee smoother transitions.

2.1.4.3 Statistical Parametric Speech Synthesis

The idea behind the US approach is to find the best fitting units and to connect them without any modifications of the signal so that no additional distortions are induced. Statistical parametric synthesis pursues a different objective which Black describes as “*generating the average of some set of similarly sounding speech segments*” [17], i.e., a natural speech database is converted into a statistical parametric model which can then be used for speech generation. In the following, the basic approach that is based on (HMMs) is introduced.

Hidden Markov Model (HMM) Synthesizer

HMM speech synthesis was first introduced by Tokuda for Japanese [18] and later adapted for the English language [19]. Figure 2.5 shows an overview of the two parts of an HMM synthesizer.

In the training part, excitation (e.g., $\log F_0$) and spectral parameters (e.g., MFCC, their delta, and delta-delta coefficients) are extracted from a given database of natural speech, similar to a speech recognizer. These parameters are then used to train context-dependent HMMs. To appropriately realize the temporal structure of a speech signal, each HMM features state duration densities that are modeled by Gaussian distributions.

For the synthesis, the desired text is first converted into a context-based label sequence. According to this sequence a sentence HMM is constructed by concatenating the trained context-dependent HMMs. From this sentence HMM excitation and spectral parameters can be derived. In conjunction with the state durations those parameters are then used to synthesize the desired text.

This creates very smooth speech which does not feature the occasional glitches of US systems. Moreover, HMM synthesis guarantees robust quality. In addition, a system’s footprint is very small and voice characteristics can be easily adjusted. However, when comparing HMM synthesizers to US systems, the overall quality

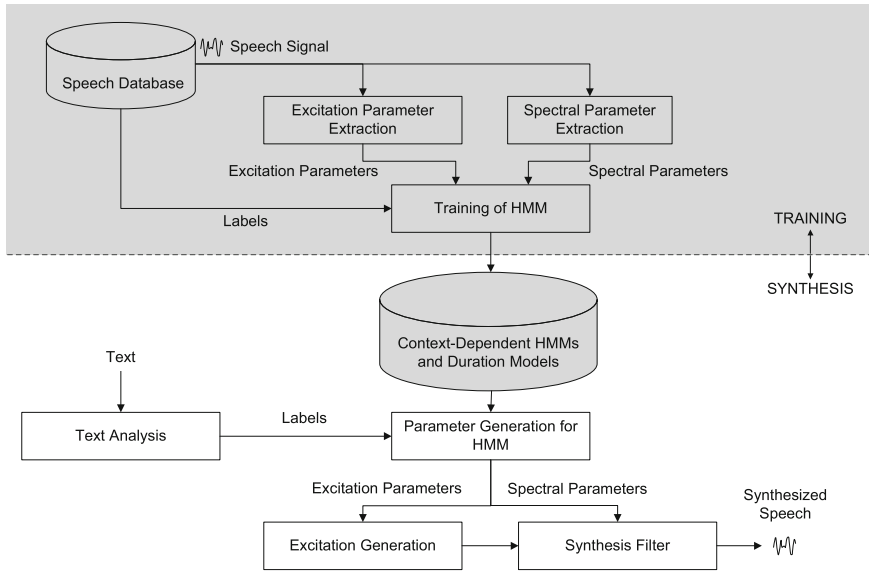


Fig. 2.5 Overview of a typical HMM based speech synthesis system [19]

still lags behind. This is mainly due to three factors: the used vocoder,³ the modeling accuracy, and an over-smoothing of excitation and spectral parameters. These impairments are responsible for the characteristic buzzy and muffled sound of speech generated by HMM synthesizers [17, 20].

2.1.4.4 Hybrid Models

In an effort to combine the positive aspects of concatenative models with those of statistical parametric synthesizers, while bypassing their downsides, different approaches on hybrid models were developed.

In [21] a system is introduced that consists of a regular HMM in the training phase. For the actual speech generation part, however, these trained HMM are then used to select an optimal phone-sized unit sequence. Therefore, this model utilizes statistical criteria for the calculation of target and join costs.

In [22] an approach on multiform segment synthesis is proposed. The used segments are either *template segments* which consist of real waveforms or *model segments* which are abstractions of speech segments produced by an HMM. The target of this algorithm is to identify the best combination of model and template segments to synthesize a given text.

³A vocoder provides a parametric representation of a speech signal.

Table 2.1 Comparison of the main characteristics of different TTS approaches

	FORMANT SYNTHESIS		PSOLA SYNTHESIS	UNIT-SELECTION SYNTHESIS	HMM SYNTHESIS
SOUND QUALITY	poor		decent	very natural	very smooth and intelligible; not very natural
VARIABILITY OF QUALITY	robust		robust	hit or miss*	robust
PERCEPTUAL IMPRESSION OF DEGRADATIONS	metallic, artificial voice		rough speech	sonic glitches, discontinuous speech	buzzy and muffled
REASONS FOR DEGRADATIONS	simplistic model		frequent concatenations of speech units; modifications of PSOLA units	rough transitions between concatenated units; lack of adequate units in the database	vocoder; modeling accuracy; over-smoothing
FOOTPRINT	very small		small	large	small
CUSTOMIZABILITY	voice characteristics can be easily adjusted		fixed to speaker of database; changes in duration and pitch are feasible	fixed to speaker and speaking style of database	voice characteristics can be easily adjusted

Note: Hybrid systems are not included in this table due to a lack of available systems.

* strong dependency on corpus and the text to be synthesized

2.1.4.5 Advantages and Disadvantages of All Approaches

As mentioned in the previous section, different approaches on speech signal generation not only lead to different levels of quality but they also feature different perceptual impressions. An overview of the characteristics of these methods is given in Table 2.1.

2.2 The Mary Text-to-Speech System (MaryTTS)

This section presents an overview of the open-source TTS system MaryTTS [23]. MaryTTS is a Java-based speech synthesis platform that is able to generate diphone, unit selection, and HMM voices. It was developed in collaboration between the DFKI's Language Technology Lab⁴ and the Institute of Phonetics⁵ at the Saarland University. The basic architecture is shown in Fig. 2.6.

MaryTTS is able to process plain input text, SABLE text⁶ and SSML⁷ text. Initially, the input has to be converted into MaryTTS's internal XML-based representation language (MaryXML). Plain text input can be directly converted, whereas SABLE and SSML have to pass through a parser before conversion. Then the tokenizer cuts the input text into tokens (i.e., words and punctuation marks) and the preprocessing unit performs a text normalization which converts numbers and abbreviations. Afterwards, a part-of-speech tagger adds word category information to each token.

From there on, two parallel branches exist: one where the prosody of the signal is modeled using GToBi,⁸ a German version of ToBi (a set of conventions for transcribing the intonation and prosodic structure of speech). The second branch, first deals with inflection endings, i.e., correct endings are assigned to ordinals and abbreviations which were identified in the preprocessing, and second, pronunciation rules for each unit are looked up in a lexicon or generated by rule, if no lexicon entry exists.

Then the prosody information and the transcriptions of the input text are merged in the phonological processing module. Here the resulting phonological representation can be restructured on the basis of phonological rules. The generated information is again added to the MaryXML structure. The following module then transforms the symbolic information into the physical domain, i.e., GToBi and a set of duration rules add information on duration and pitch to the MaryXML structure.

⁴DFKI Language Technology Lab: <http://www.dfki.de/lt/>, last accessed 22.04.2016.

⁵Institute of Phonetics, Saarland University: <http://www.coli.uni-saarland.de/groups/WB/Phonetics/>, last accessed 22.04.2016.

⁶SABLE: https://www.cs.cmu.edu/~awb/festival_demos/sable.html, last accessed 21.04.2016.

⁷SSML: <https://www.w3.org/TR/speech-synthesis/>, last accessed 21.04.2016.

⁸GToBi: http://www.gtobi.uni-koeln.de/gm_gtobi_modell.html, last accessed 22.04.2016.

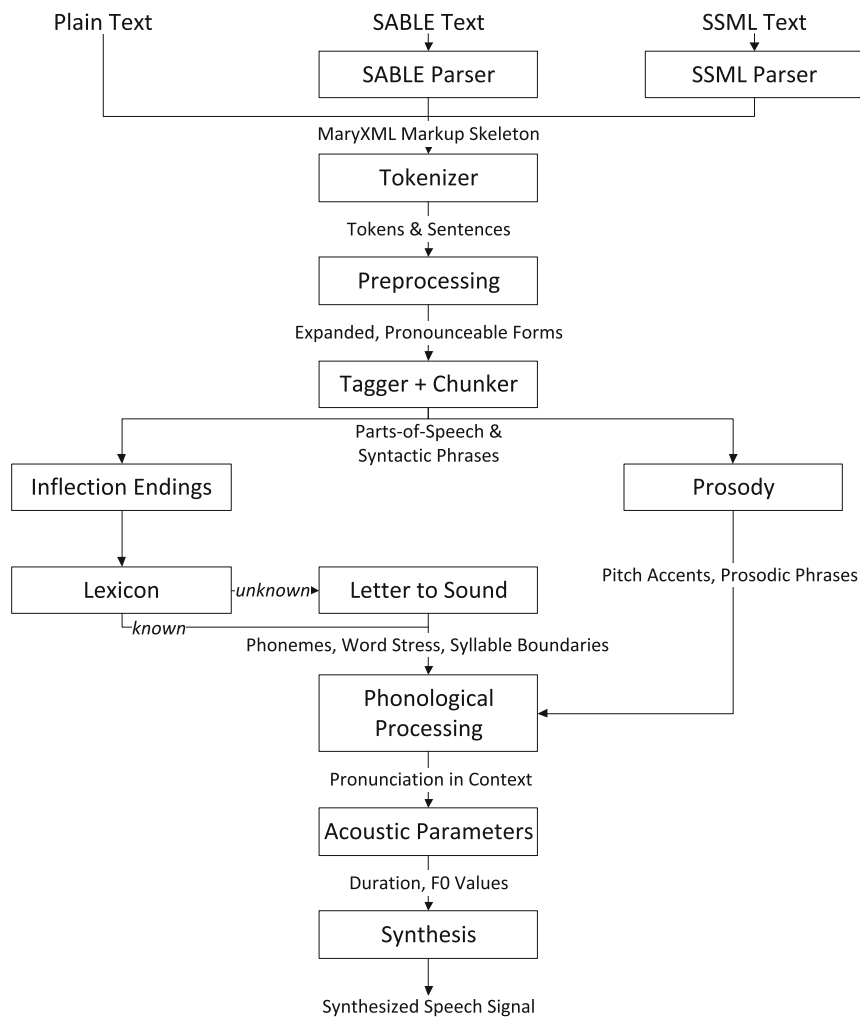


Fig. 2.6 Processing architecture within MaryTTS [23, 24]

Finally, the synthesis module utilizes the updated MaryXML information to generate a synthesized speech file. Therefore, an MBROLA based synthesizer can be applied to generate a diphone voice and more recent versions also allow the generation of unit selection as well as HMM voices.

References

1. Vary P, Heute U, Hess W (1998) Sprachsynthese. Digitale Sprachsignalverarbeitung. B.G. Teubner, Stuttgart, pp 465–497
2. Pfister B, Kaufmann T (2008) Sprachverarbeitung - Grundlagen und Methoden der Sprachsynthese und Spracherkennung. Springer, New York
3. Fujisaki H (1981) Dynamic Characteristics of Voice Fundamental Frequency in Speech and Singing. *Acoust Anal Physiol Interpret STL-QPSR* 22:1–20
4. Fujisaki H (2004) Information, Prosody, and Modeling with Emphasis on Tonal Features of Speech. *Speech Prosody* 23:1–10
5. Flanagan JL, Ishizaka K, Shipley KL (1975) Synthesis of Speech from a Dynamic Model of the Vocal Cords and Vocal Tract. *BELL Syst Tech J* 54(3):485–506
6. Klatt DH (1987) Review of Text-To-Speech Conversion for English. *J Acoust Soc Am* 82:737–793
7. Klatt DH (1980) Software for a Cascade/Parallel Formant Synthesizer. *J Acoust Soc Am* 67:971–995
8. Huang X, Acero A, Hon H-W (2001) Spoken Language Processing - A Guide to Theory, Algorithm, and System Development. Prentice Hall PTR, New Jersey
9. Taylor PA (2009) Text-To-Speech Synthesis. Cambridge University Press, Cambridge
10. Ley J (2016) Blind programmieren - Wenn der Computer schneller spricht, als ein Sehender hört. <http://www.golem.de/news/blind-programmieren-wenn-der-computer-schneller-spricht-als-ein-sehender-hoert-1601-117767.html>. Accessed 01 Nov 2016
11. Moos A, Trouvain J (2007) Comprehension of Ultra-Fast Speech - Blind versus Normally Hearing Persons. In: Proceedings of the 16th International Congress of Phonetic Sciences, Saarbrücken, pp 677–680
12. Institut für Kommunikationsforschung und Phonetik, Universität Bonn. Projekt MiLCA – Gesprochene Sprache. https://web.archive.org/web/20070613001637/http://www.ikp.uni-bonn.de/dt/lehre/Milca/mmk/content/mmk_s322.xhtml. Accessed 10 Feb 2015
13. Black A, Taylor PA (1994) CHATR: A Generic Speech Synthesis System. In: Proceedings of the 15th International Conference on Computational Linguistics (COLING), Kyoto, Japan, pp 983–986
14. Campbell NW (1996) CHATR: A High-Definition Speech Re-Sequencing System. In: Proceedings of the 3rd ASA/ASJ Joint Meeting, Hawaii, USA, pp 1223–1228
15. Hunt AJ, Black AW (1996) Unit Selection in a Concatenative Speech Synthesis System Using a Large Speech Database. In: Proceedings of the 21st International Conference on Acoustics, Speech and Signal Processing, Atlanta, Georgia, USA, pp 373–376
16. Campbell NW, Black A (1996) Progress in Speech Synthesis. Prosody and the Selection of Source Units for Concatenation Synthesis. Springer, New York, pp 279–291
17. Black A, Zen H, Tokuda K (2007) Statistical Parametric Speech Synthesis. In: Proceedings of the IEEE International Conference of Acoustics, Speech and Processing (ICASSP), Honolulu, Hawaii, USA, pp 1229–1232
18. Tokuda K, Yoshimura T, Masuko T, Kobayashi T, Kitamura T (2000) Speech Parameter Generation Algorithms for HMM-base Speech Synthesis. In: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Istanbul, Turkey, pp 1315–1318
19. Tokuda K, Zen H, Black AW (2002) An HMM-Based Speech Synthesis System Applied to English. In: Proceedings of 2002 IEEE Speech Synthesis Workshop (SSW), Santa Monica, USA, pp 227–230
20. Zen H, Tokuda K, Black A (2009) Statistical Parametric Speech Synthesis. In: Speech Communication, vol 51. Elsevier Science Publishers BV, pp 1039–1064
21. Ling Z-H, Qin L, Lu H, Gao Y, Dai L-R, Wang R-H, Jiang Y, Zhao Z-W, Yang J-H, Chen J, Hu G-P (2007) The USTC and iFlytek Speech Synthesis Systems for Blizzard Challenge 2007. In: Proceedings of the 3rd Blizzard Workshop in conjunction with the 6th ISCA Workshop on Speech Synthesis (SSW), Bonn, Germany

22. Pollet V, Breen A (2008) Synthesis by Generation and Concatenation of Multiform Segments. In: Proceedings of the 9th Annual Conference of the International Speech Communication Association (Interspeech 2011), Brisbane, Australia, pp 1825–1828
23. Schröder M, Trouvain J (2001) The German Text-to-Speech Synthesis System MARY: A Tool for Research, Development and Teaching. In: Proceedings of the 4th ISCA Workshop on Speech Synthesis
24. DFKI GmbH. Mary Text-to-Speech: Architecture Walkthrough, 20 April 2016. <http://mary.dfki.de/documentation/module-architecture.html>

Quality of Synthetic Speech
Perceptual Dimensions, Influencing Factors, and
Instrumental Assessment

Hinterleitner, F.

2017, XVI, 157 p. 29 illus., Hardcover

ISBN: 978-981-10-3733-7