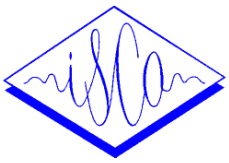

3rd European Conference on
Speech Communication and Technology
EUROSPEECH'93
Berlin, Germany, September 19-23, 1993

SYNTACTIC PROCESSING AND PROSODY CONTROL IN THE SVOX TTS SYSTEM FOR GERMAN

Christof Traber

Speech and Language Processing Group, Electronics Laboratory,
Swiss Federal Institute of Technology (ETH), CH-8092 Zurich, Switzerland
e-mail: traber@ife.ethz.ch

ABSTRACT

An overview of the current state of the SVOX TTS system for German is presented in this article. A detailed presentation of the text analysis component is given and it is shown how word and sentence structure analysis and even number-to-phoneme conversion and grapheme-to-phoneme conversion can be expressed concisely and in an easily readable way in a definite-clause-grammar (DCG) formalism, which is interpreted by a chart-parser. We also give an overview of the prosody control in the system, which consists of rule-based accentuation and prosodic phrasing and of a purely statistical and neural-network-based acoustic interpretation of suprasegmental phonological information.

Keywords: *text-to-speech synthesis, prosody control, syntactic analysis, grapheme-to-phoneme conversion*

10.21437/Eurospeech.1993-470

1. INTRODUCTION

The goal of the TTS project at our laboratory was to build a flexible TTS research tool rather than a small real-time TTS system. However, during the last year the first and very bulky system has been partly re-implemented and speeded up in order to prepare the system for an application in the near future (reading aloud weather forecasts over telephone line). It is also planned to use the system as part of an automatic dialog application. This article presents the current state of the TTS system.

The first part of the system, called "transcription", comprises the morpho-syntactic text analysis and the generation of a phonological representation of each utterance, i.e., a minimal, yet segmentally and prosodically sufficient speaker-independent representation of the utterance. This representation consists of phoneme strings, accent markers of different degrees and phrase boundaries of different strengths. The transcription can be viewed as the "competence" component of the system, and the processing methods are of knowledge- and rule-based character.

The second part, called "phono-acoustical" model, maps the phonological information onto acoustic data and can be viewed as the "performance" component. It consists mainly of F_0 and duration control and of diphone concatenation for the generation of the speech signal /1/. Originally, LP-coded diphones at 10 kS/s were used, but we are also experimenting with new speech production methods (e.g., the PSOLA methods /2/). The control of the prosodic parameters was achieved by applying statistical methods with the aim to imitate the natural prosody of a trained reference speaker, speaking in a "neutral" style.

The following sections describe the syntactic and prosodic components in more detail. We will mainly focus on the syntactic processing in this article. Due to lack of space, prosody control is only summarized and references are given to articles that explain the present prosody control in much more detail.

2. SYNTACTIC PROCESSING

2.1. Overview

By speaking of syntactic processing in this article we refer to all processes of our TTS system that involve syntactic parsing using grammars and lexicons. Syntactic processing in our system includes word and sentence structure analysis, but also the derivation of the pronunciation of numbers and abbreviations, and even grapheme-to-phoneme conversion.

2.2. Text "preprocessing"

Many TTS systems carry out some text preprocessing as a separate first step, in which abbreviations are expanded to their full forms and digit sequences are converted into the appropriate pronunciation. However, in languages like German the proper pronunciation of abbreviations and numbers may depend on the surrounding syntactic constellation. For example, "der 3. Oktober" must be read as "der dritte Oktober" (nominative) whereas "am 3. Oktober" is expanded to "am dritten Oktober" (dative). In our view, such expansions should therefore not be done before but together with the syntactic sentence analysis. The solution we adopted for this problem is shown in section 2.4.

Text preprocessing in our TTS system is currently a simple application-specific text formatting only.

2.3. Word analysis and sentence analysis

The task of the word analysis is to assign a phonetic transcription and part-of-speech information and other morpho-syntactic features such as case, number, and gender (which may be ambiguous) to each orthographic word. This is achieved by first looking up the word in a full-form lexicon, which mainly contains function words and adverbs. If no full-form entry can be found, the word is decomposed into morphemes which are stored in a morpheme lexicon. The morphological decomposition is driven by a word grammar,

and parsing is done using a chart parser.

The sentence analysis uses the results of the word analysis to find a sentence parse according to a sentence grammar.

In addition to word grammar and sentence grammar, full-form lexicon and morpheme lexicon there exists a set of (bi-directional) rules which handle some regular allomorphic (graphemic and phonetic) variations (like, e.g., devoicing of stem-final consonants or insertions and deletions of certain phonemes and graphemes at particular morpheme boundaries) /3/.

The word and sentence analysis framework was designed by our former colleague Thomas Russi, and the grammars were originally implemented as unification-based transition networks (UTN) /3/. After a recent re-implementation of the chart parser the grammars have been converted automatically into the equally powerful, but in our view somewhat more concise and more easily readable and maintainable definite-clause-grammar (DCG) formalism /4/.

Some examples of possible full-form lexicon entries in the present formalism are

```
das   ARTB(nom sg neu) ['das] !preposition
das   REL(acc neu)     ['das] !relative pronoun
bald  ADV(temp)        ['balt] !adverb
```

and examples of morpheme-lexicon entries are

```
spielA+ VS(conj1)      ['ʃpi:lA+]!verb stem
t#      VE(conj1 sg3 pres)  [t#]!verb ending
lich+   ASUFF()            [lɪç+]!adj. suffix
```

All lexicon entries consist of their graphemic and phonetic representation, the word or morpheme category and additional information like case, number, gender, inflection class etc. The special symbols appearing in the morpheme lexicon are used by the afore-mentioned allomorphic rules. "!" separates entries from comments. Upper and lower case letters are currently considered equal throughout the syntactic processing.

Examples of possible sentence grammar rules are

```
S() -->
  NP(nom ?Number)  VP(?Number) * 1 10
NP(?Ca ?Nu) -->
  ARTB (?Ca ?Nu ?Ge) N(?Ca ?Nu ?Ge) * 1 5
```

(The rules of the word grammar are of the same form.) The first rule states that a sentence may be composed of a noun phrase in nominative case and a verb phrase, and both phrases must agree in the number feature. The second rule states that a noun phrase may be composed of a definite article and a noun which agree in case, number, and gender. Feature terms starting with "?" are variables, all other feature terms are constants (atoms). "*" terminates the list of subconstituents. The number after "*" is a flag that controls whether this rule will appear as a node in the syntactic tree, and the last number is a penalty value. These values are used when building the optimal parse tree: the nodes of the tree inherit the penalty values of the corresponding grammar rules, and the parse tree with globally minimal summed penalty is chosen as the final result of the sentence analysis.

The penalty values are also used when no full-sentence parse can be found, in which case an artificial sentence tree is constructed by combining constituents that have been found during chart parsing and which constitute the optimal path through the chart (i.e., the path with minimal penalty). A bottom-up strategy is applied in all parsings in order to derive as many useful constituents as possible. Thus, in the case that no full parse can be found the sentence analysis is automatically and smoothly changed into a phrase-level parser.

2.4. Analysis of numbers and abbreviations

Numbers, i.e., digit sequences, are treated like ordinary words, the "morphemes" of which are single digits or digit sequences with a corresponding pronunciation. A special grammar for cardinal and ordinal numbers (which is actually part of the regular word grammar) defines well-formed combinations of these elements and thereby the mapping from numbers onto pronunciations (and also vice versa). Since this number grammar yields all possible pronunciations and the corresponding syntactic information (e.g., case) the sentence grammar chooses the appropriate form depending on the wider context. This automatically solves problems of the kind stated in section 2.2.

Abbreviations consisting of one word only (like German "bzw." for "beziehungsweise") can simply be stored in the full-form lexicon in graphemic and phonetic form. In the case of two- or more-word abbreviations (like "z. B." for "zum Beispiel" and "z. Z." for "zur Zeit", in which "z." represents different words) the pronunciations of the individual items must be defined by lexicon entries and the connection of the two or more parts is defined by a rule in the sentence grammar which ensures that only parts belonging together may actually be combined.

2. 5. Fully declarative grapheme-to-phoneme conversion

Until recently, our system did not contain any grapheme-to-phoneme conversion (or letter-to-sound rules) at all, since the idea was to have a fully lexicon-based system. However, even the largest lexicons do not comprise all possible words, especially proper names may be missing /5/. It is therefore desirable to have an additional grapheme-to-phoneme conversion in a TTS system. There exist different approaches to grapheme-to-phoneme conversion, including for instance the classical ordered set of rewrite rules, e.g., /6/, rhyming /5/ and analogy /7/ methods.

Since we were looking for a grapheme-to-phoneme (GTP) conversion that would fit nicely into our present morphological analysis and which should also work if only a part of a word is missing, we have found a quite uncommon scheme of GTP conversion (or rather GTP mapping), namely a fully declarative, structure-oriented GTP conversion by means of a grammar for German and foreign word stems and a lexicon of consonant and vowel clusters. In analogy to the word grammar, which defines legal decompositions of words into lexically stored morphemes, this stem grammar (which is actually simply an extension of the regular word grammar) defines legal decompositions of stems into lexically stored graphemic and phonetic units. In other words, the stem grammar is a declarative formulation of the graphotactic and phonotactic rules of the German language. Although developed independently and for a somewhat different purpose, this approach to GTP conversion shares much of the spirit of the YorkTalk structural phonology approach /8/. Our method also bears some similarity to the "flexible GPC rules" described in /9/.

To illustrate the interplay between morphological analysis and GTP conversion we sketch a highly simplified word grammar for German nouns (lexical categories are written with lower case letters):

Regular morphology

```
N(?Cas ?Num) -->
 REP_ESTEM() ESTEM() nend(?Cas ?Num)* 1 1

ESTEM() -->
  REP_PREF() STEM() REP_SUFF() * 1 1

REP_ESTEM() --> * 0 0
REP_ESTEM() --> ESTEM() REP_ESTEM() * 0 1

REP_PREF() --> * 0 0
REP_PREF() --> pref() REP_PREF() * 0 1

REP_SUFF() --> * 0 0
REP_SUFF() --> suff() REP_SUFF() * 0 1
```

Stem analysis

```
STEM() --> lex_stem() * 1 1! known lexical stem
STEM() --> GSTEM() * 1 100 ! new german stem
STEM() --> FSTEM() * 1 200 ! new foreign stem

GSTEM() --> icc() long_vow() fcc1() * 1 1
GSTEM() --> icc() short_vow() fcc2()* 1 1
GSTEM() --> icc() spec_vow() fcc1() * 1 1
GSTEM() --> icc() spec_vow() * 1 5

FSTEM() --> REP_SYL() * 1 1
REP_SYL() --> SYL() * 0 0
REP_SYL() --> SYL() REP_SYL() * 0 1
etc.
```

These rules state that a noun is in general composed of a repetition of stems (possibly surrounded by prefixes and suffixes) and a noun inflection ending at the end. The stems may either be lexically given (lex_stem) or be unknown German or foreign stems. German stems are basically analyzed as an initial consonant cluster, a vowel or vowel cluster and a final consonant cluster, which may be empty after a special vowel cluster. Foreign stems are analyzed as syllable sequences (not further developed here). The high penalty in the rules for the unknown stems ensures that a lexically given stem is preferred over a stem decomposed by rules. The actual grapheme-to-phoneme mapping is done by supplying a lexicon of consonant and vowel clusters, e.g.,

```
st icc()      ['ʃt]   ah   spec_vow() [a:]
m  icc()       ['m]   ee   spec_vow() [e:]
i  long_vow() [i:]    l    fcc1()      [l]
a  long_vow() [a:]    lt   fcc2()      [lt]
a  short_vow() [a]    rsch fcc2()      [rʃ]
```

Our GTP conversion is not fully elaborate yet and we cannot give any accuracy figures here. Figure 1 shows the result of sentence, word and stem analysis for the sentence "wir beschreiben eine Schnellzugfahrt" ("we describe a ride on a fast train"), in which the stems "schnell", and "zug" are unknown but correctly converted into the phonetic form by the rules described above (the phonetic transcriptions of words and morphemes are shown as leaves of the tree in an ASCII-coded form).

The following properties characterize our GTP approach:

- The transition from the regular lexicon-based morphological analysis to the application of GTP conversion is extremely smooth, since all is done within the very same framework and formalism.
- GTP conversion is basically implemented by writing a grammar that describes all possible graphotactic and phonotactic structures of German and foreign words and stems; this results in a very concise and easily readable GTP conversion without too much interaction between different rules; in our system this proved to be the easy part; the difficulties lie mostly in choosing appropriate penalties for all rules in order to always get the desired optimal pronunciation.
- It is well possible to access very wide structural contexts which may influence the phonetic transcription of graphemic elements (e.g., the choice of stressed/unstressed first syllable of a word depending on a the kind of a suffix at the end); influences of narrow contexts are mostly captured in the consonant and vowel cluster lexicon.
- Morphological decomposition of words, stress assignment, the choice of the appropriate set of GTP conversion rules (German/foreign) and stem boundary detection based on consonant cluster analysis is all done at the same time; this eliminates the problems with classical rewrite rules stated in /10,p.772/ (order of stress prediction and phoneme prediction, left-to-right or right-to-left processing).

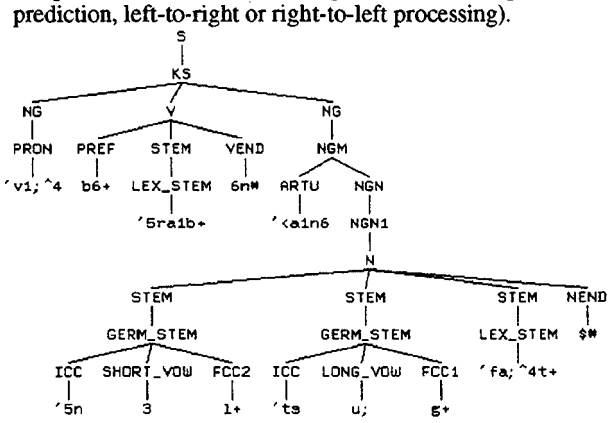

Figure 1: Syntactic tree produced by our TTS system; explanation see text.

3. PROSODY CONTROL

3.1. Accentuation and prosodic phrasing

In our system, accents are derived from the syntactic tree structure of a sentence (like the one shown in the preceding section) by applying two main rules:

1) some word categories (mostly function words) are declared to be unstressed; all other words initially receive a primary accent (accent level 1) on their word stress position
2) the so-called nuclear stress rule is applied in cyclic fashion from the leaves of the syntactic tree to the root, thereby weighting accents against each other /11,12/; the nuclear stress rule states that each syntactic constituent has a nucleus (e.g., the noun within a noun phrase), which remains primary-accented, whereas all other accents within the same constituent are reduced (i.e., increased in numeric value); in our system, a collection of syntactic subtree patterns define the nucleus position of each possible syntactic constituent

The result of this accentuation procedure is a sentence in which each syllable carries a certain accent value. These values are then modified again after the prosodic phrasing.

The phrasing algorithm that is used in our system is a slightly modified version of an algorithm described in /13/, which works in the following steps:

1) An initial boundary is set between each pair of neighboring words; the strength of this boundary is set to the level of the common ancestor node of both words in the syntactic tree (i.e., the stronger the syntactic connection between words, the less the initial separation strength)
2) unaccented (clitic) words are enclosed in that neighboring phrase from which they are less separated; this is achieved by deleting the weaker one of the boundaries to the left and right of the unaccented word
3) boundaries are deleted further in the order of their strength (from weak boundaries to stronger ones) under the condition that the number of accents and syllables in the resulting phrase does not exceed a certain value (rhythmic melting of neighboring temporary phrases); otherwise the boundary remains

The result of this algorithm is a sentence split up into syntactically motivated phrases.

The application of the nuclear stress rule in the accentuation procedure may lead to unnaturally many accentuation levels which are not phonetically meaningful (it is impossible to distinguish more than three or four accent levels in speech). The initial accentuation is therefore modified again after the phrasing of the sentence has been determined: within each phrase, all accent strengths are increased as much as possible, but the original hierarchical ordering is maintained; this ensures that relative prominences of syllables within a phrase remain as predicted by the accentuation rules and that each phrase receives a primary accent (the phrase accent), while keeping the accent values in a reasonable range. Accents, phrase boundaries and phonetic transcriptions of all words together constitute the phonological representation of an utterance.

3.2. Duration control and F_0 control

Duration values for all segments of the artificial speech signal are generated based on different factors, such as the kind of the current segment (fricative, sonorant, ...), the kind of the surrounding segments, accentuation, position of the segment within the syllable, the foot, the phrase, and the sentence. These factors, which are known to influence the segmental duration, can easily be derived from the phonological representation of an utterance. The duration values in our system are generated from a binary coding of these factors and combinations thereof by applying a so-called generalized linear model. In this statistical model, the output value is a linear function of the input values. The coefficients used in this linear function are parameters that were statistically estimated such that the model optimally predicts the segmental durations of a set of given natural sentences. More details can be found in /14/.

In our system, the most successful fundamental frequency control was achieved using a recurrent neural network (described in detail in /15/). Basically, a neural network can simply be regarded as a non-linear statistical model with a large number of parameters. These parameters (called weights) are estimated in a training procedure such that they optimally predict a set of given outputs from the corresponding inputs. This approach to F_0 generation is therefore closely related to the approach taken for duration control. In our system, F_0 contours are generated syllable-wise. For each syllable of the utterance, the accent value of the concerned syllable and the accent values of a number of surrounding syllables are fed into the network together with some attributes describing segmental properties of the syllable as well as the position of the syllable within the phrase and the sentence. The output of the network is the F_0 pattern for the current syllable (represented by 8 samples of the F_0 contour). The concatenation of all syllable patterns yields the F_0 contour of the whole utterance.

4. CONCLUSION

We have been able to implement all text analysis in our TTS system, including syntactic, morphological and grapheme-to-phoneme analysis within a pure, concise and aesthetically satisfying DCG framework. This text analysis can be used for text-to-phoneme conversion as well as for phoneme-to-text conversion. In future, the morpheme lexicon will be enlarged, and grapheme-to-phoneme conversion will be completed.

An "engineering" approach to the generation of prosodic parameters has been chosen by imitating the natural prosody of a particular speaker using statistical methods. This is a rather "inexpensive" approach to prosody control, but it provides little insight into the underlying phonetics.

ACKNOWLEDGEMENTS

I would like to thank the Swiss PTT for funding this work. I am very much indebted to Susanne Elisabeth Werner for implementing the first version of the grapheme-to-phoneme conversion.

REFERENCES

/1/: Kaeslin, H. (1986). A systematic approach to the extraction of diphone elements from natural speech. *IEEE Transactions on Acoustics, Speech, and Signal Processing (ASSP), 34,* 264-271.

/2/: Moulines, E., & Charpentier, F. (1990). Pitch-synchronous waveform processing techniques for text-to-speech synthesis using diphones. *Speech Communication, 9,* 453-467.

/3/: Russi, T. (1992). A framework for morphological and syntactic analysis and its application in a text-to-speech system for German. In G. Bailly & C. Benoît, eds., *Talking Machines: Theories, Models, and Designs,* 163-182. North-Holland.

/4/: Pereira, F. C. N., & Warren, D. H. D. (1980). Definite Clause Grammars for Language Analysis – A Survey of the Formalism and a Comparison with Augmented Transition Networks. *Artificial Intelligence, 13,* 231-278.

/5/: Coker, C. H., Church, K. W., & Liberman, M. Y. (1990) Morphology and Rhyming: Two Powerful Alternatives to Letter-to-Sound Rules for Speech Synthesis. In *Proceedings of the ESCA Workshop on Speech Synthesis,* Autrans, France, 83-86.

/6/: Allen, J., Hunnicutt, M. S., & Klatt, D. (1987). *From text to speech: The MITalk system.* Cambridge University Press.

/7/: Dedina, M. J., & Nusbaum, H. C. (1991). PRONOUNCE: a program for pronunciation by analogy. *Computer Speech and Language, 5,* 55-64.

/8/: Coleman, J. (1992). "Synthesis-by-rule" without segments or rewrite-rules. In G. Bailly & C. Benoît, eds., *Talking Machines: Theories, Models, and Designs,* 43-57. North-Holland.

/9/: Sullivan, K. P. H., & Damper, R. I. (1992). Novel-word pronunciation within a text-to-speech system. In G. Bailly & C. Benoît, eds., *Talking Machines: Theories, Models, and Designs,* 183-195. North-Holland.

/10/: Klatt, D. H. (1987). Review of text-to-speech conversion for English. *Journal of the Acoustical Society of America, 82,* 737-793.

/11/: Kiparsky, P. (1966). Über den deutschen Akzent. *Studia Grammatica VII,* Akademie-Verlag, Berlin.

/12/: Selkirk, E. O. (1984). *Phonology and Syntax: The Relation between Sound and Structure.* MIT Press, Cambridge, Mass.

/13/: Bierwisch, M. (1966). Regeln für die Intonation deutscher Sätze. *Studia Grammatica VII,* Akademie-Verlag, Berlin.

/14/: Huber, K. (1990). A statistical model of duration control for speech synthesis. In *Proceedings of the 5th European Signal Processing Conference (EUSIPCO),* Barcelona, 1127-1130.

/15/: Traber, C. (1992). F_0 generation with a database of natural F_0 patterns and with a neural network. In G. Bailly & C. Benoît, eds., *Talking Machines: Theories, Models, and Designs,* 287-304. North-Holland.