# A TEXT-TO-SPEECH SYSTEM BASED ENTIRELY ON RULES

Rolf Carlson and Björn Granström, Dept. of Speech Communication, KTH, S-100 44 Stockholm 70, Sweden

## Summary

When reading a text a native speaker pronounces most words correctly even if they are unknown to him. During this process he makes use of his knowledge of the language, the semantic content and the syntax. However, if we take away all information except the spelling and some pronunciation rules on the word level, the task would be more difficult. This is basically the case in our text-to-speech synthesis system containing neither semantic and syntactic analysis nor a word or morpheme dictionary. At the conference the function of our present synthesis system will be discussed. The result shows that such a system might well be based on rules rather than on an extensive dictionary. Furthermore a useful tool in speech synthesis work is described (i.e. a programming language).

## Introduction

In the present paper some general aspects on text-to-speech systems will be discussed. Furthermore, a system for Swedish that is in progress will be described, as well as a useful tool in this work (i.e. a programming language).

The purpose for developing a text-to-speech system could be two-fold. First to answer the question: is it possible to make such a system at all? Is it possible to formulate our present knowledge in terms of well-specified rules? In what sense does the "output" deviate from a natural utterance and why? Is the deviation perceptually important?

The second, but not less important argument for working on this subject is of course the need for talking machines for different kinds of applications as, for example, reading machines for the blind. [1, 9] However, even if these two goals not are contradicting the emphasis on the work could be made in different ways resulting in different kinds of simplifications. Let us assume that we try to develop a strategy to predict emphasis. This strategy does probably generate errors and the output will be misleading in some cases. For an application the unemphasized version might be the practical one at this point but the faults are important and interesting and lead us to research areas so far neglected and regarded as uninteresting. The work on how to handle "new" information in a sentence, not earlier mentioned in a passage, is a typical example of this kind.[1] We could regard a synthesis system as a part of a functional model of the transformation of an abstract message via the acoustical waveform to a perceived message. The perception is important in this process since it guides us to the crucial events that carry the information. It is our belief that this link so far has been payed too little attention.

The general speech synthesis system consists of a sequence of transformations each reflecting a piece of our knowledge of the speech process. We could, when heading for an application jump into this process at any point depending on the structure of the input and the wanted degree of sophistication or simplicity. The total system may guide the general work on speech to important subjects and force the researchers to formulate the gained knowledge.

## A programming language for rule specified transformations

As mentioned in the introduction there is a need for a notation in which linguistic and phonetic knowledge could be formulated. The system as seen in for example Sound Pattern of English [7] is useful but restricted. Some important generalizations have to be made, especially to cover the mathematics of continuous variables like duration and intonation. It should be noted that the framework presented below is not especially made for speech synthesis, even if it is handy in this work, but could be used for formulating models of different kinds in the speech and and language research. The notation has been implemented as a computer language [5] and used in different kinds of applications.

The model to be tested has to be formulated as a sequence of ordered rules. The rules work on string elements and these elements must be defined by the user. The definition can include a specification of distinctive features and variables associated with a string character. The basic features could be labeled according to the user's choice and they can be specified as plus or minus or unspecified giving, in fact, a ternary opposition. The basic features can also be grouped into any number of secondary features giving a handy description of natural classes. The variables used must also be labeled and they can be either one- or two-dimensional. The two-dimensional ones can have more than one value associated with an element. The latter kind of variables are typically intended for time/value or space/value description. There is no inherent restriction on the number of possible variables.

The basic structure of our rules is:

(1) $X \longrightarrow Y / A \ \& \ B$

where & marks the place where the structural description X occurs in the context description A B. Y denotes the structural change and $\longrightarrow$, and / are mere delimiters. In (1) A, B, X, Y stand for strings that could all be empty. When A and B are empty we simply write

(2) $X \longrightarrow Y$

i.e. a context-free rule (compare rule (6) below). X being empty implies an insertion and when Y is empty a deletion takes place.

The strings A, B, X are strings of elements and could contain string characters like a and i in (3) but alsc more weakly or more strictly specified elements within < > parentheses. Such a description referring to strings like apî, así could be

(3) a < +CONS, -VOICE > < i, STRESS > 3, x:=F1 >

The < > parentheses could thus contain variable and feature conditions on the context and also one string character. Here x:=F1 means that the external variable x is given the value of the parameter F1 in this element and could be used in the structural change. The inclusion of optional elements in the string makes compact rule descriptions possible. In this formalism such an operation is designated by a ( , ) notation, e.g.

(4) < +CONS > (1,3)

specifying the length of a consonant cluster of minimally one and maximally three segments. A blank or zero in the first position means none and in the second position a blank means an infinite number.

The structural change Y in (1) - (2) is a string of elements similar to the strings X, A, B but here the < > parentheses express the result of a change like

(5) < a, +ACCENT, F1=N >

An element will thus be changed to one corresponding to the character "a"; the feature +ACCENT will be added to its feature specifications and the value of the variable F1 will be altered to N.

If a value is to be inserted, a ":=" is used instead of a "=" and if a value is to be deleted a "=#" is used. N is a numerical expression of arbitrary complexity where internal variables from the element itself and external variables as x in (3) could be used with standard numerical functions.

Special system rules could be added to the rule system giving the operator the possibility to examine the result of rule applications at different stages in the processing of the string. At any point in the rule sequence comments (C....) and break points (BREAK) could be inserted. When a BREAK occurs the operator can examine the status of the currently modified string as well as the status of the program and rules. A special routine could also be used at this point both for listening tests and for plotting of graphs.

By means of a joy stick external variables could be introduced by the operator at certain points. The joy stick coordinates are displayed on a screen for visual feedback. These variables could be used in the following rules resulting in any kind of change. This method has been used in several experiments, one of them concerning perception of duration. [6]

Each time a rule is applied a rule-specific counter is incremented. By this method the productivity of each rule can be statistically evaluated.

## General aspects on text-to-speech systems

In our present work on a text-to-speech system some views on the reading process have been guiding. We know that a speaker normally pronounces a spelled word correctly, even if it is unknown to him. This may be done by some analogy or similarity process in which he compares the unknown word to memorized lexical items. However, such a process could be formulated as the general pronunciation rules that exist in most languages.

In a computer program for speech synthesis of an unlimited vocabulary, the two basically different methods could be thought of: a lexical one as opposed to a rule-based one. However, the data base in the first case will be overwhelming unless the words are not stored as morphs. Especially in Swedish, where compounds are extremely common, the use of morphs is the only possible way. At this point we already have to introduce rules (e.g. rules for compounding). In a similar way pronunciation changes of morphs, because of affixes, have to be formulated. Furthermore, when a "new" word appears, which seems to be very probable despite the use of a rather huge data base, we want it to be pronounced as correctly as possible. This could only be done with the help of pronunciation rules.

To conclude, we feel that each general system has to contain rules and if these reflect a reality in the language in question, the lexicon could be reduced in a drastic way. In our present system we try to get along without any lexicon at all. Of course the system will generate faults but if these may be done by an uneducated speaker as well, the listener might accept them without misunderstanding the message. Furthermore, the system will in this way illustrate the present knowledge of how the language is pronounced.

## A text-to-speech system for Swedish

### Transformations to a phonemic representation

At a first glance the transformation from graphic to phonemic representation seems to be easier for Swedish than for English. This is true if phenomena like vowel quality and quantity, as well as accent, is neglected. Unfortunately, this could not be done and the difficulties appear to be equal, though of different kinds.

In Swedish we have to find the primary stressed syllable in the morph and there may also exist a secondary stressed syllable. During this search for stressed syllables, different cues in the spelling must be used. Simple indicators like double-spelling as well as certain endings (e.g. -ent) are obvious. We find that some vowels are likely to carry stress (e.g. Å, Ä, and Ö), while others (e.g. A and E) are not. Certain consonant combinations indicate a "heavy" syllable while others do not. The general philosophy in our system is that obvious stress marking and consonant change appear in the beginning of the process. More sophisticated rules are applied later depending on which transformations that have been active before that point.

Morph-boundary insertions will appear during the whole process and are very important for the accent settings. If a compound is found the primary stressed syllable in the non-initial morph of the word is reduced to a secondary or zero stress although the quality and quantity of the vowel is kept unchanged. A secondary stressed vowel as a part of a root-morph behaves in a different way.

### Transformation of a phonetic to an acoustic representation

Despite how the phonetic representation has been developed, phonological and phonetic rules have to be applied to derive a phonetic representation which could be transformed to a synthesized acoustic wave.

These rules work primarily on the segmental level and have a relatively close contextual description. The prosodic rules are exceptions in this case. [3,4]

One example is a segment duration rule

$$(6)\quad < \text{SEGMENT} > \rightarrow < \text{DURATION} = T*(A+B+C)* \text{EXP}(-\text{LOG}(B)*0.12-\text{LOG}(A)*0.35) >$$

where T is a nominal duration and A, B are variables depending on the position and "word" or phrase length.

Work on the perception of duration is now in progress in order to specify the framework in which the descriptive model should fit. [6]

In the phonological component rules for devoicing, flapping and so forth are easily included since phenomena like that are well understood.

If, as in our case, a terminal synthesizer is used coarticulation and reduction have to be specified on a parametric level. This is artificial in some sense, and a model of the vocal tract with natural restrictions should solve many of these problems.

The synthesizer mentioned is an OVE III [10] supplied with an external voice-source. [11] This voice source is a functional model but has the advantage of using parameters closely related to physiology.

### Work in progress on the text-to-speech system

Work is now under way to optimize the text-to-speech system at different levels. This is done by corrections, insertions, and reordering of rules. Test sequences of different kinds like VCV lists and short stories are used in listening tests. For testing the graph to phoneme transformation we use the most common 10.000 words in Swedish. [2] When the words have been processed the transcriptions are automatically compared to stored "correct" transcriptions

with help of a computer program. The work on the system has been simplified to a high degree by this automatic procedure. It should be noted that even if an "uncorrect" transcription is given, many words are accepted by listeners, especially in running synthesized speech.

## Final comments

Some important parts of the reading process have been neglected in the present paper. Semantics as well as syntax have not been dealt with in a proper way, since they have been beyond the scope of our presentation as well as our functional model at present. However, at least some parts of these aspects could and should be included in the future.

## References


1. J. Allen: "Speech synthesis from unrestricted text", pp. 416-428 in Speech Synthesis (eds. J. L. Flanagan and L. R. Rabiner), Dowden, Hutchinson & Ross, Inc., Stroudsburg, Penn. 1973.
2. S. Allén: Nusvensk frekvensordbok, 1 (Frequency Dictionary of Present-Day Swedish), Almqvist & Wiksell, Stockholm 1970.
3. R. Carlson and B. Granström: "Word accent, emphatic stress, and syntax in a synthesis by rule scheme for Swedish", STL-QPSR 2-3/1973, pp. 31-36.
4. R. Carlson, Y. Erikson, B. Granström, B. Lindblom, and K. Rapp: "Studies of the rhythm and intonation of Swedish", Speech Communication, Vol. 2; Almqvist & Wiksell, Stockholm 1975.
5. R. Carlson and B. Granström: "A phonetically oriented programming language for rule description of speech", Speech Communication, Vol. 2; to be publ. by Almqvist & Wiksell, Stockholm 1975.
6. R. Carlson and B. Granström: "Perception of segmental duration", to be publ. in Structure and Process in Speech Perception (eds. A. Cohen, S. G. Nooteboom), Springer-Verlag, Berlin 1975.
7. N. Chomsky and M. Halle: The Sound Pattern of English, Narper & Row, Publ, New York 1968.
8. C. H. Coker, N. Umeda, and C. P. Browman: "Automatic synthesis from ordinary English text", pp. 400-411 in Speech Synthesis (eds. J. L. Flanagan and L. R. Rabiner), Dowden, Hutchinson & Ross, Inc., Stroudsburg, Penn. 1973.
9. F. S. Cooper, J. H. Gaitenby, I. G. Mattingly, P. W. Nye, and G. N. Sholes: "Audible outputs of a reading machine for the blind", Haskins Lab., SR-29/30 (1972), pp. 91-95.
10. J. Liljencrants: "The OVE III speech synthesizer", IEEE Transac. on Audio and Electroacoustics, Vol. AU-16, March 1968, pp. 137-140.
11. M. Rothenberg, R. Carlson, B. Granström, and J. Gauffin: "A three-parameter voice source for speech synthesis". Speech Communication, Vol. 2; Almqvist & Wiksell, Stockholm 1975.