First International Conference on
Spoken Language Processing (ICSLP 90)
Kobe, Japan
November 18-22, 1990

TETOS - A TEXT-TO-SPEECH SYSTEM FOR GERMAN

Klaus Wothke

IBM Germany
Heidelberg Scientific Center
Tiergartenstr. 15, D-69 Heidelberg

ABSTRACT

A text-to-speech system was developed which reads aloud arbitrary German texts. It converts an orthographic input text into synthetic speech in 3 main steps: At first the words of the text are preprocessed and abbreviations, special characters, and digits are replaced by their full orthographic correlates. In a second step the preprocessed words are phonetically transcribed by means of about 1,370 letter-to-phone rules. The rate of incorrect transcriptions is near 2% of the running words of a text. Finally the phone symbols in the phonetic transcriptions are mapped on control codes for a commercial speech synthesizer. The control codes are sent to the synthesizer where they trigger the production of synthetic speech.

I. INTRODUCTION

This paper deals with the results of a project carried out between 1985 and 1988. The aims of the project were:

- Development of a text-to-speech system which reads aloud unrestricted German text with good acoustic quality.
- Design of a system architecture which allows an easy modification of the linguistic knowledge of the system.
- Separation of the software from the language specific data, in order to permit the use of the software for other languages.

The outcome of the project is the text-to-speech system TETOS.

II. ENVIRONMENT OF TETOS

The environment of TETOS is illustrated in fig. 1. TETOS has 4 input files:

- A dictionary containing abbreviations and special characters with their full orthographic correlates.
- A file with letter-to-phone rules.
- A file with mappings of phone symbols on control codes for the synthesizer.
- An arbitrary orthographic German text which is to be converted into speech.

By means of the linguistic knowledge in the first 3 files TETOS maps an arbitrary orthographic German text on its phonetic transcription, which subsequently is mapped on a sequence of control codes for the commercial speech synthesizer SynPhonix 200.

Each of the 3 files exists in 2 representations: An external representation, which permits an easy modification of the

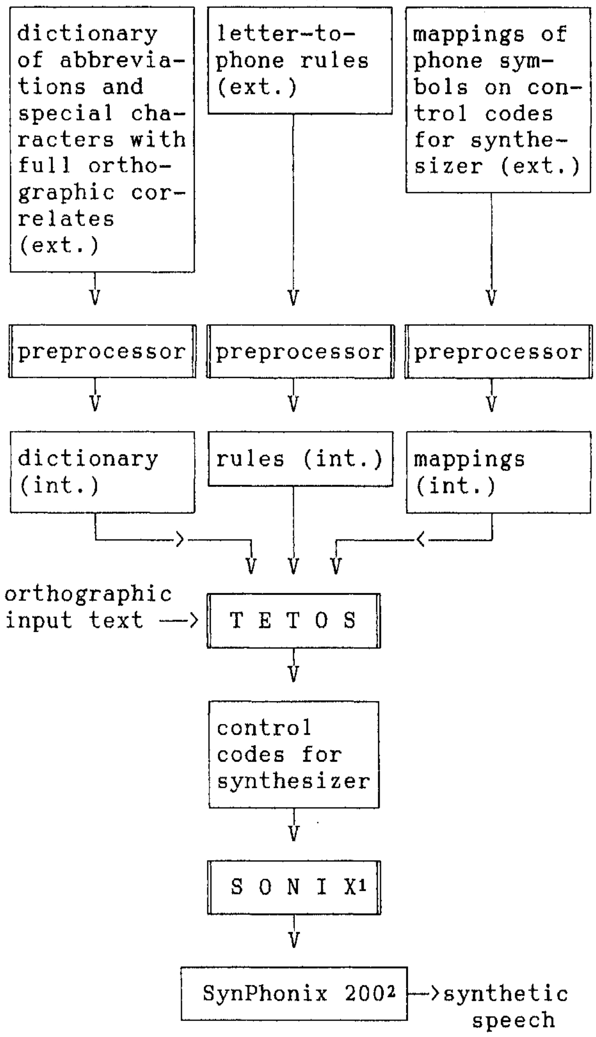

Fig.1. Environment of the text-to-speech system TETOS.

1 SONIX is a trademark of Artic Technologies.
2 SynPhonix 200 is a trademark of Artic Technologies.

linguistic knowledge by means of an editor, and an internal representation which is no longer human readable, but which is more suitable for the use of the linguistic knowledge by TETOS. The internal representations are generated from the external representations by preprocessors, which also check the external representations for formal errors.

Almost all language specific data are in the first 3 input files. Only the knowledge necessary for the conversion of digits to their full orthographic representation (cf. 3.2) had to be integrated into program code. In order to adapt TETOS to other languages it is only necessary to fill the 3 files with the language specific data and to replace the program module for the conversion of digits by the corresponding language specific module.

Each time TETOS generates a sequence of control codes for an orthographic word, it sends this sequence to SONIX, which is an interface between the synthesizer SynPhonix 200 and application software such as TETOS. With the control codes SONIX triggers the generation of speech by the synthesizer.

TETOS and the preprocessors are written in the programming language Turbo C[1]. The system runs under MS DOS[2].

III. GERMAN LANGUAGE SPECIFIC KNOWLEDGE

3.1 Abbreviations and Special Characters

Before abbreviations and special characters are phonetically transcribed and mapped on control codes for the synthesizer, they are replaced by their full orthographic correlates. E.g. '%' is replaced by 'Prozent', 'ccm' by 'kubikzentimeter'. For this process TETOS uses a dictionary of about 1,800 abbreviations and special characters with their orthographic correlates. These are

- the special characters of the ASCII table (e.g. '@', '$', '&', '%').
- the most common abbreviations out of those listed in [3].
- additional common abbreviations found in newspaper texts, directions for use, travel folders, scientific papers etc.

3.2 Digits

Like abbreviations and special characters also digits are replaced by their full orthographic correlates, before they are phonetically transcribed and mapped on control codes. The replacement is performed by a program module which contains as an integrated part the German specific knowledge about the replacement process. The module first subclassifies the digits into the 4 classes of whole cardinal numbers, real numbers, dates, and others. Depending on the classification the full orthographic correlate is generated. Examples:

The cardinal number '21' is replaced by 'einundzwanzig'.

The real number '47,21' is replaced by 'siebenundvierzig komma zwei eins'.

The orthographic correlates of dates are inflected. The type of inflection is highly predictable with the aid of the preceding word. E.g.: '12.7.1988' is replaced by

- 'zwölfte siebte neunzehnhundertachtundachtzig', if 'der' precedes.
- 'zwölften siebten neunzehnhundertachtundachtzig', if one of the words 'am', 'dem', 'den', 'des', 'vom', 'zum' precedes.
- 'zwölfter siebter neunzehnhundertachtundachtzig', else.

All other digits (which are not cardinal numbers, real numbers, or dates) are replaced by an orthographic representation which leads to their literal pronunciation. E.g. '1.3' is mapped on 'eins punkt drei'

3.3 Letter-to-Phone Rules

By means of the letter-to-phone rules TETOS maps all orthographic strings on their phonetic transcription. The letter-to-phone rules are applied to orthographic strings which either occur literally in the original text or which are the full orthographic correlates of abbreviations, special characters, or digits.

Currently about 1,370 letter-to-phone rules are used. They were created in the following way:

- We started with a set of about 200 rules created by A. Günther (published in [1], pp. 316).
- This initial set was tested against the entries in the German pronunciation dictionary "Großes Wörterbuch der deutschen Aussprache" [2] and iteratively improved.
- A pronunciation dictionary does not contain any inflected words; therefore the transcription of inflectional endings was tested by applying the rules to German texts. This resulted in further improvement of the rules.

The formalism for the formulation of the rules (= external representation) consists of the following elements:

Comments: Each line starting with the string

'+' (but not: '+SET ')

is considered as a comment and may contain any information.

1 Turbo C is a trademark of Borland International, Inc.

2 MS DOS is a trademark of the Microsoft Corporation.

Set Definitions: A collection of letter strings which are often used together in the rules may be combined into a set which is designated by a set identifier. Instead of the concrete letter strings the set identifier may then be used in the rules. – A set definition has the general format

+SET X={x_1 x_2 ... x_n} + ...

where
- X is the set identifier (a capital letter).
- each x_i is a string of lower case letters.
- '+ ...' is a comment (optional).

Rules: A rule has the general format

X[Y]Z=>W + ...

Read: Replace the letter string Y by the phone string W, if Y occurs right of the letter pattern X and left of the letter pattern Z. ('+ ...' is an optional comment.)

Some important conventions:
- X and Z may contain letter strings as well as set identifiers.
- In X and Z the characters @ and # may be used: @ is the wildcard character and represents the set of all possible characters. # represents the word boundary.
- A set identifier (including the wildcard) may be followed by the restricted Kleene operator. Its format is *n ($1 \leq n \leq 5$). For arbitrary n A*n always implies A*0. Example: 'A*2' means 'AA', 'A', and ''. Thus the following 2 rule formulations are functionally equivalent

```
                              bAA[a]B=>a:
bA*2[a]B=>a:     <====>      bA[a]B=>a:
                              b[a]B=>a:
```

- The rules are ordered. This means: for a given letter string their application is tried in their order in the rule file.
- Object language strings are in lower case letters. Capital letters are set identifiers.

To illustrate, how letter-to-phone rules can be expressed in the rule formalism, an excerpt from the German phoneticization rules for the letter 't' is listed below:

```
+---------------------------------------+
+LETTERS FOR BACK VOWELS:               |
+SET B={a o u}                          |
+---------------------------------------+
+CONSONANTAL LETTERS:                   |
+SET C={b c ch d f g ... z β}           |
+---------------------------------------+
+NOMINAL ENDINGS:                       |
+SET E={e em en es er ern ... ste sten} |
+---------------------------------------+
+ 't'-RULES:                            |
i[t]ät=>'t        + 'individualität' etc.|
[tie]llE*1#=>tsjϵ + 'essentiell' etc.   |
k[t]ie=>ts        + only: '-aktie/n-'   |
@[t]ien=>ts       + 'kroatien' etc.     |
[t]ii=>ts         + 'initiieren' etc.   |
[ti]B=>tsj        + 'exerzitium' etc.   |
[thek]=>'te:k     + 'discothek' etc.    |
[t]her#=>t        + 'dorther' etc.      |
[th]=>t           + STANDARD            |
C[tt]=>tt         + 'bartträger' etc.   |
[tth]=>t          + 'matthias' etc.     |
[tt]=>t           + STANDARD            |
[tz]=>ts          + STANDARD            |
[t]=>t            + STANDARD            |
+---------------------------------------+
```

3.4 Mappings of Phone Symbols on Synthesizer Control Codes

When the phonetic transcription of an orthographic word has been generated, it is mapped on a sequence of synthesizer control codes. For this task TETOS accesses a file containing the control codes for each phone symbol. The control codes determine the phone to be synthesized and they control its duration, pitch, rate, amplitude, and formant transitions. Currently the file contains 76 (partially context sensitive) mappings for phone symbols and punctuation marks. – Punctuation marks are not phonetically transcribed. Instead, they are directly mapped on control codes for long or short pause.

IV. INTERNAL MACRO STRUCTURE OF TETOS

When TETOS is started, it first loads the internal representation
- of the dictionary of abbreviations and special characters with their full orthographic correlates,
- of the letter-to-phone rules,
- of the mappings of phone symbols and punctuation marks on synthesizer control codes.

Then the orthographic input text is processed line by line. Each line is split into segments. Segments are:
- Punctuation marks.
- Letter strings, which may contain embedded '.' or '.-' (e.g. 'z.B' or 'röm.-kath').
- Digit strings, which may contain embedded '.', ',', or ':' (e.g. '12.10.1950', '3.2.2', '21,03', '12:30').
- Special characters (e.g. '%', '&').

- Strings of blanks or other characters not covered by the other types of segments.

How the process of text-to-speech conversion continues, depends on the segment type:

- A punctuation mark is directly mapped on a control code which triggers a pause.
- Letter strings are looked up in the dictionary of abbreviations and special characters. If they are found there, they are replaced by their full orthographic correlate. Then the letter-to-phone rules are applied and the resulting phonetic transcription is mapped on a sequence of control codes.
- Digits are first subclassified into cardinal numbers, real numbers, dates, and others. Depending on their class they are replaced by their full orthographic correlate. Then the letter-to-phone rules are applied and the resulting phonetic transcription is mapped on a sequence of control codes.
- Special characters are looked up in the dictionary of abbreviations and special characters. If they are found there, they are replaced by their full orthographic correlate which in turn is phonetically transcribed and mapped on a sequence of control codes. If a special character is not found in the dictionary, it is not converted into speech; instead TETOS directly processes the following segment.
- Strings of blanks and other characters are not converted into speech. Each time such a segment occurs TETOS directly processes the following segment.

When TETOS has mapped a segment on control codes, it waits until the control codes of the preceding segment have been processed by the synthesizer, before it sends the new control codes to the synthesizer. Then the next text segment is processed.

V. CONCLUSIONS

TETOS converts unrestricted German texts into well intelligible synthetic speech. About 6% of the different words and only about 2% of the running words of a text contain an incorrect letter-to-phone conversion, but most of these erroneous transcriptions do not lead to a prominent deviation in the synthetic speech.

The transcription errors are mainly a consequence of the fact that the system does not have any knowledge of the morph structure of the words. In German the phonetic transcription of a letter is often not only dependent on the letter itself and on its orthographic context, but also on its morphological context. The absence of information about the morph structure causes e.g. errors of the following type: When the letters 'b', 'd', 'g' occur in a compound word at the end of a morph and in front of a lexical morph or a prefix beginning with a vocalic letter or a liquid, they are incorrectly mapped on the voiced plosives [b], [d], [g] instead of [p], [t], [k], respectively. Furthermore, the glottal stop symbol is missing if the initial letter of the following morph is a vocalic letter. Example: 'Mittagessen' (engl.: lunch) is incorrectly transcribed as ['mita:gɛsn]* instead ['mita:k?ɛsn]. - A German text-to-speech system which takes into account morph boundaries is RELEXIS [4].

A further reason for the transcription errors is that TETOS does not currently take into account the prosodic features. - Word stress is taken into account to a certain degree during the phonetic transcription, but it is ignored during the subsequent mapping on control codes.

Two very important features of TETOS result from the fact that most of the linguistic knowledge is separate from the software and that this knowledge is represented in human readable and mnemotechnically favourable formalisms: The linguistic knowledge can be modified easily without changing the software and TETOS can be adapted to other languages without any modifications of the software. (Only exception is the replacement of digits by their orthographic correlates, which had to be integrated into program code.)

REFERENCES

[1] Bierfert, H. (1985): Automatische Spracherkennung - von der Theorie zur Praxis. Tübingen.

[2] Krech, E.-M. et al. (eds.) (1982): Großes Wörterbuch der deutschen Aussprache. Leipzig.

[3] Werlin, J. (1987): Duden. Wörterbuch der Abkürzungen. 3., rev. ed. Mannheim etc.

[4] Wolf, H.E. / Strecker, F. / Fries, G. (1989): Text-Sprache-Umsetzung für Anwendungen bei automatischen Informations- und Transaktionssystemen. In: Informationstechnik 31(1989)5. PP. 334-341.