2008 IEEE International Conference on Automation, Quality and Testing, Robotics

Proceedings - TOME III

AQTR 2008
THETA 16

May 22-25 2008
Cluj-Napoca
Romania

IEEE

Institute of Electrical and Electronics Engineers

tttc

IEEE Computer Society - Test Technology Technical Council

Technical University of Cluj-Napoca, Romania, Department of Automation

IPA SA CLUJ CIFATT

IPA - R&D Institute for Automation Center for Technology Transfer Cluj-Napoca, Romania

Voice Synthesis Application based on Syllable Concatenation

O. Buza[1], G.I. Toderean[1], J. Domokos[2], A. Zs. Bodo[3]
[1] Technical University of Cluj, Ovidiu.Buza@com.utcluj.ro, Gavril.Toderean@com.utcluj.ro
[2] Sapientia University of Tirgu Mures, jdomokos@com.utcluj.ro
[3] Technical University of Cluj, zsolt.bodo@gmail.com

Abstract: - **This article presents a voice synthesis application based on syllable concatenation. The system is dedicated for Romanian language, so it was need to work on special rules to decompose Romanian text into syllables. Also for preserving initial prosody of text, accentuation of syllables inside word had to be determined. Then we have recorded a vocal database with the most frequent syllables of Romanian language. A unit matching algorithm matches linguistic units from the input text and acoustic units from database. Acoustic units are then concatenated and converted into sound by mean of a synthesizer.**

I. Introduction

Concatenation of waveforms produces high level of naturalness in speech outcome. Syllable-based method is a particular case of corpus methods and presents the advantage of high level of quality and low cost of database maintaining. And like corpus approaches, syllable-based methods induce less concatenation errors because of small number of concatenation points inside the unit sequence.

We consider that syllable approach is very appropriate in Romanian case, because Romanian spoken language contains a big number of opened vowels that gives a special rhythm of speech in which syllables are easy to detect.

Our text-to-speech system is organized on five modules: linguistic analyse module, prosodic analyse module, database management module, unit matching module and speech synthesis module (fig.1).

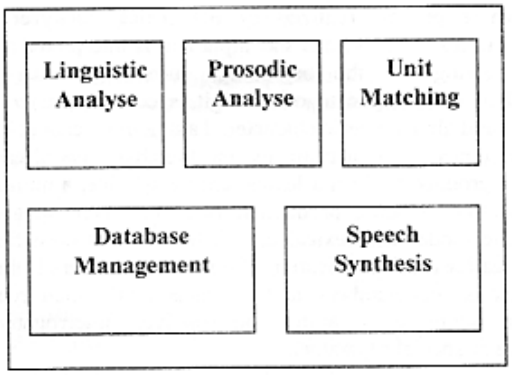

Figure 1. Main modules of our Text-to-Speech system

Linguistic analyze module makes input text analyze and extracts phonetic basic units, that are the syllables.

Prosody analyse module intend to detect segmental prosody units based on input text. In this stage, position of stressed syllable inside a word has been detected.

Database management module makes all database-connected operations. Vocal database contains a subset of Romanian language syllables and was obtained by recording voice of a male speaker.

Unit matching module assures matching between linguistic units from text and acoustic units from database. Matching has to be optimized, as not all the existent syllables are recorded in vocal database.

Speech synthesis module realizes concatenation of waveforms and generates speech based on this acoustic units sequence.

About implementation of the system, first we have built a linguistic analyzer module that was capable to split the input text into syllables. Next step was to determine accentuation by mean of a phonetic analyzer. Then we have automatically produced a database with PCM coded syllables of Romanian language.

Synthesis was done by concatenating acoustic units from database and giving appropriate commands to the computer sound blaster for voice generation.

II. Text Analysis

First stage in text analysis is the detection of linguistic units: sentences, words and segmental units that in our approach are the word syllables.

Detection of sentences and words is done based on punctuation and literal separators. For detection of syllables we had to design a set of linguistic rules for splitting words into syllables, inspired from Romanian syntax rules ([2], [3]).

The principle used in detecting linguistic units is illustrated in fig. 2. Here we can see the structure of text analyser that corresponds to four modules designed for detection of units, prosody information and unit processing.

These modules are: lexical analysis module for detection of basic units, phonetic analysis module for generating prosody information, high level analysis module for detection of high-level units, and processing shell -for unit processing.

1-4244-2576-1/08/S20.00 ©2008 IEEE

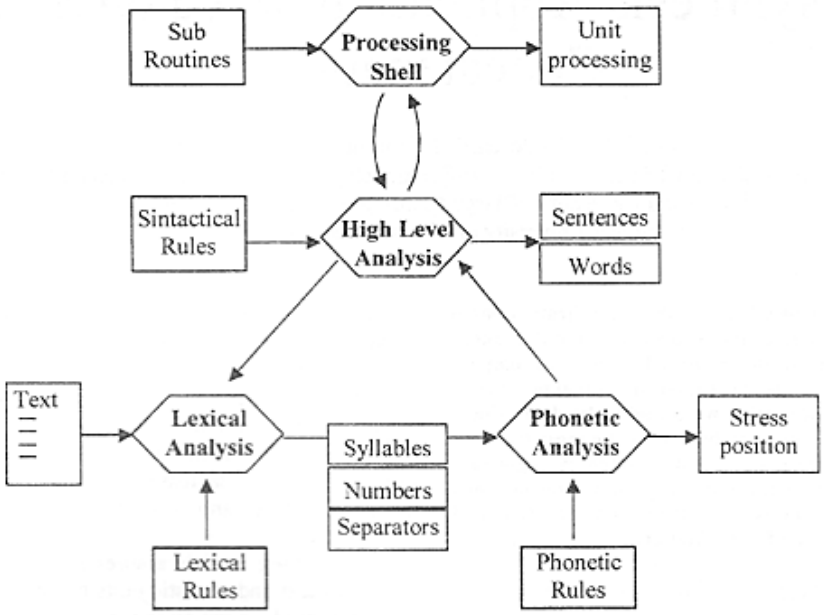

Figure 2. Text analyser for syllable detection

Processing shell accomplishes the unit processing task and controls the subsequent modules. The shell calls high-level analyser for returning main syntactic units. High-level analyser calls the lexical analyser for input text parsing and detection of basic units. Then phonetic analysis module is called for generating stress information.

Lexical analyser extracts text characters and clusters them into basic units. We refer to the detection of alphabetical characters, numerical characters, special characters and punctuation marks. Using special lexical rules (that have been presented in [8]), alphabetical characters are clustered as syllables, digits are clustered as numbers and special characters and punctuation marks are used in determining of word and sentence boundaries.

Phonetic analyser gets the syllables between two breaking characters and detects stress position, i.e. the accentuated syllable from corresponding word.

Then, high-level analyser takes the syllables, special characters and numbers provided by the lexical analyser, and also prosodic information, and constructs high-level units: words and sentences. Also basic sentence verification is done here.

Processing shell finally takes linguistic units provided from the previous levels and, based on some computing subroutines, classifies and stores them in appropriate structures. From here synthesis module will construct the acoustic waves and will synthesize the text.

III. Lexical Analysis for Syllable Detection

Lexical analyzer is called by the higher level modules for detection of basic lexical units: syllables, breaking characters and numbers. The lexical analyzer is made by using LEX scanner generator [4]. LEX generates a lexical scanner starting from an input grammar that describes the parsing rules. Grammar is written in BNF standard form and specifies character sequences that can be recognized from the input. These sequences refer to syllables, special characters, separators and numbers. Also BNF grammar specifies the actions to be taken in the response of input matching, actions that will be accomplished by the processing shell subroutines.

The whole process realized by the lexical analyzer is illustrated in fig. 3. As we can see, input text is interpreted as a character string. At the beginning, current character is classified in following categories: digit, special character or separator, and alphanumeric character. Taking into account left and right context, current character and the characters already parsed are grouped to form a lexical unit: a syllable, a number or a separator. Specific production rules for each category indicate the mode each lexical unit is formed and classified, and also realize a subclasification of units (for numbers if they are integer or real numbers, and for separators – their type: word or sentence separator, affirmative, interrogative, imperative or special separator).

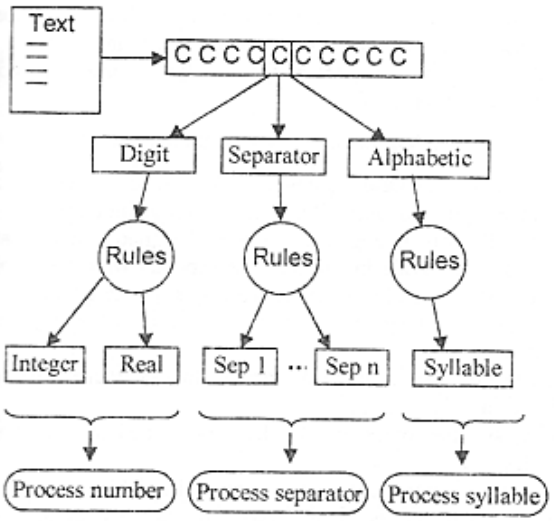

Figure 3. Lexical analyser for syllable detection

Once the unit type and subtype is identified, corresponding character sequence is stored and transmitted to the high-level analyzer by mean of specific actions, as they will be described in next paragraph (*Process syllable, Process number, Process separator*).

A. Specific actions of lexical analyser

Specific actions inform high-level module about matching of syllables, numbers and breaking characters. Inside lexical parser three types of input response actions are defined as follows:

1) *Process syllable* – this is the action to be taken when a syllable is matched in specific location of one word.

Special attention is taken when a syllable is matched at the beginning of a word. In Romanian, different word decomposition rules apply when a character sequence occurs at the beginning or in the middle or the final part of a word.

2) *Process number* – is the action to be taken when a number is matched from the input. The number is identified as INTEGER or REAL type. In future stage, numbers will be translated in orthographic alphabetical form.

3) *Process separator* - is the action corresponding to a breaking character matching from the input. Breaking characters and punctuation marks are used for detecting word and sentence boundaries.

B. Syllable rules matching

Regarding syllable rules matching process inside lexical analyser, two types of rule sets were made: a basic set consisting of three general rules, and a large set of exception rules which statues the exceptions from the basic set.

1) *Basic set* shows the general decomposition rules for Romanian. It consists from three rules:

```
syllable={CONS}*{VOC}                          (R1)
syllable={CONS}*{VOC}{CONS}/{CONS}             (R2)
syllable={CONS}*{VOC}{CONS}*/{SEP}             (R3)
```

First rule says that a syllable consists of a sequence of consonants followed by a vowel.

Second rule statues that a syllable can be finished by a consonant if the beginning of the next syllable is also a consonant.

Third rule says that one or more consonants can be placed at the final part of a syllable if this is the last syllable of a word.

2) *The exception set* is made up from the rules that are exceptions from the three rules of above. These exceptions are situated in the front of basic rules. If no rule from the exception set is matched, then the syllable is treated by the basic rules. At this time, the exception set is made up by more then 100 rules. Rules are grouped in subsets that refer to resembling character sequences. All these rules are explained in [7], [8].

IV. Syllable Accentuation

The principle for determining syllable accentuation is shown in fig. 4.

The parser returns current word from input stream. The word consists of series of phonemes F_1, F_2, ..., F_k and is delimited by a separator *S*. Phonetic analyser reads this word and detects syllable accentuation based on phonetic rules.

In Romanian, stressed syllable can be one of last four syllables of the word: S_n, S_{n-1}, S_{n-2} or S_{n-3}, (S_n is the last syllable). Most often, stress is placed at last but one position.

The rules set consists of this general rule (S_{n-1} syllable is stressed):

```
{LIT}+/{SEP}        { return(SN_1);}
```

and a consistent set of exceptions, organized in classes of words having the same termination. In [7] one can find the complete set of rules.

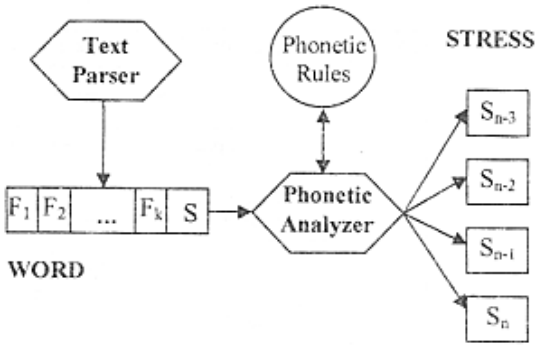

Figure 4. The principle of detecting syllable accentuation

V. HIGH LEVEL ANALYSIS

High-level analyser takes low-level information: syllables, special characters and numbers from the lexical analysis module and constructs high-level structures: words and sentences.

High-level analysis module takes a regular production rules set which specifies the syntax of input text. Input text is considered to be a set of sentences, each sentence – a set of words and each word is composed by one, two or more syllables. Sentences, and words respectively, are bounded by separators.

Hierarchical structure of high-level units is presented in fig. 5. In this diagram, greyed units (syllables, numbers and separators) are the outcome of lexical analyser. Thereby, high-level or syntactical analyser module invokes lexical analyser for providing next lexical unit from input text: a syllable, a number or a separator.

Based on the lexical units, at this level are formed syntactical units as words, sentences and text. High-level analyser also has the capability to call, for each syntactical unit separated from text, a specific subroutine from processing shell module.

In our implementation, words and sentences are processed by calling two subroutines from processing shell: *Process_Word* and *Process_Sentence*.

Based on corresponding terminators, at this level, sentences are classified as regular, imperative or interrogative. Such a classification is very important for modifying speech prosody (in future developments).

VI. VOCAL DATABASE AND UNIT MATCHING

Vocal database contains PCM coded waveform of Romanian syllables. Actually, vocal database includes only a subset of Romanian language syllables, designed as a tree data structure. Each node in the tree corresponds with a syllable characteristic, and a leaf represents the appropriate syllable.

Units have been inserted in database following this classification:

- after length of syllables : two, three or four character syllables and also singular phonemes;
- after syllable place inside the word: median and final syllables;
- after accentuation: accentuated or normal syllables.

Unit matching is done according to this three–layer classification. If one syllable is not founded in vocal database, this will be constructed from other syllables and separate phonemes that are also recorded. Following situations may appear:

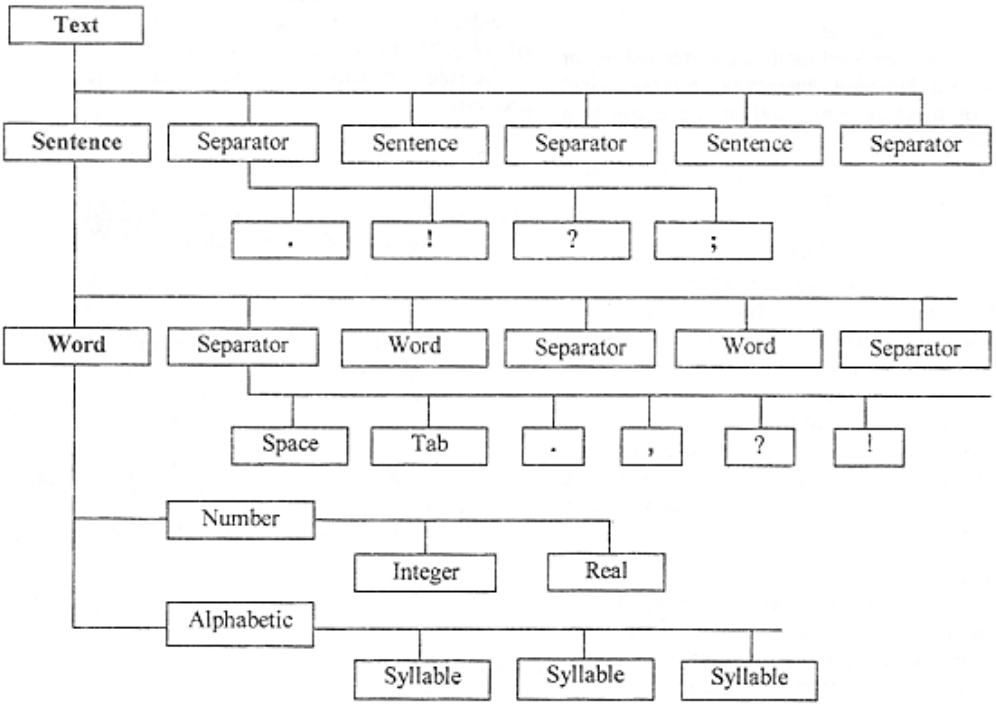

Figure 5. Hierarchical structure of high-level analysis

(a) Syllable is matched in appropriate accentuated form. In this case acoustic unit will be directly used for concatenation.

(b) Syllable is matched but not the accentuation. In this case, unit is reconstructed from other syllables and phonemes which abide by the necessary accentuation.

(c) Syllable is not matched, so it will be constructed from existing syllables and phonemes.

VII. Implementation

System implementation is illustrated in fig. 6.

First stage in implementing the voice synthesis system was building of vocal database. In our case, vocal database was recorded from a male speech using a standard acquisition system. For the two-character syllables, speaker recorded pseudo-words that contained each necessary syllable as middle-syllable and some of them as final syllable. Pseudo-words were designed for normal syllables as well as for stressed syllables. Also we have done a statistic of the most used three-character syllables in Romanian and corresponding words were recorded.

After recording of audio sequences, a normalization stage was required. This stage was accomplished through a general audio processing program, which assisted us to normalize speech signal in pitch and amplitude.

To extract acoustic units from recorded normalized speech, the segmentation stage has followed. We have used a signal processing tool which can detect the signal boundaries for each phoneme present at the input. Boundaries were manually adjusted and the corresponding signal for each syllable was cut out and saved in the database file.

At this moment, audio database contains 562 syllables: 386 two-character syllables (accentuated or not, 283 middle-word syllables and 103 ending-word syllables), 139 most frequent three-character syllables and 37 four-character syllables. Syllables that are not included in database are synthesized from those existing and separate phonemes that are also recorded.

VIII. Conclusions and results

We have presented in this article our voice synthesis application based on syllable concatenation. Must be mentioned that special efforts have been done to accomplish the text processing stage. After serious researches in linguistic field, we have designed one set of rules for detecting word syllables and a second set for determining which syllable is accentuated in each word. Even these sets are not complete, they cover yet a good majority of cases. The lexical analyzer is entirely based on rules that assure more than 85% correct syllable detection at this moment, since accentuation analyser provides about 75% correct detection rate.

The advantages of detecting syllables through a rules-driven analyser are: separation between syllables detection and system code (different from [9], where syllables detection algorithm is integrated in source code); from here we have easy readability and accessibility of rules. Other authors ([1]) have used LEX only for pre-processing stage of text analysis, and not for units detection process itself. Some methods support only a restricted domain ([6]), since our method supports all Romanian vocabulary. The rules-driven method also needs fewer resources than dictionary-based methods (like [5]).

About speech synthesis outcome, first results are encouraging, and after a post-recording stage of syllable normalization we have obtained a good quality of speech synthesis. In future implementations, we consider that using a run-time adaptive correction in concatenation points will smooth the output signal and improve the system performance.

References

[1] D. Burileanu, et al., "A Parser-Based Text Preprocessor for Romanian Language TTS Synthesis", *Proceedings of EUROSPEECH'99*, Budapest, Hungary, vol. 5, pp. 2063-2066, Sep. 1999.

[2] G. Constantinescu-Dobridor, *Sintaxa limbii române*, Editura Ştiinţifică, Bucureşti, 1994

[3] G. Ciompec et al., *Limba română contemporană. Fonetică, fonologie, morfologie*, Editura Didactică şi Pedagogică, Bucureşti, 1985

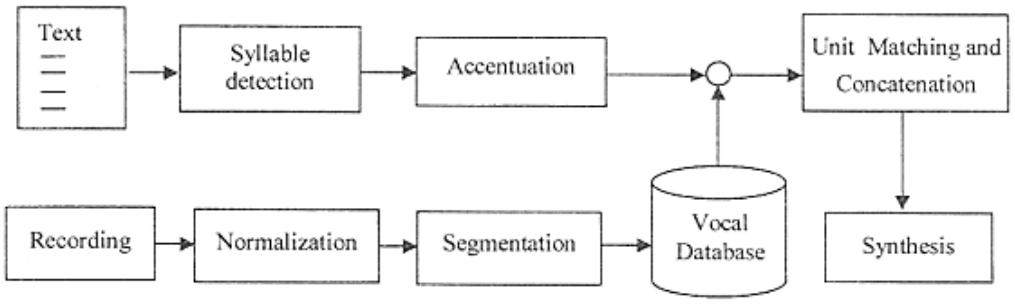

Figure 6. System implementation

[4] Free Software Foundation, "Flex - a scanner generator", http://www.gnu.org /software/flex/manual, October 2005.
[5] A. Hunt, A. Black, "Unit selection in a concatenative speech synthesis system using a large speech database", *Proc. ICASSP '96*, Atlanta, GA, May 1996, pp. 373–376.
[6] E. Lewis, M. Tatham, "Word And Syllable Concatenation In Text-To-Speech Synthesis", *Sixth European Conference on Speech Communications and Technology*, pages 615—618, ESCA, September 1999.
[7] O. Buza, *Vocal interractive systems*, doctoral paper, Electronics and Telecommunications Faculty, Technical University of Cluj-Napoca, 2005
[8] O. Buza, G. Toderean, "Syllable detection for Romanian text-to-speech synthesis", *Sixth International Conference on Communications* COMM'06 Bucarest, June 2006, pp. 135-138.
[9] C. Burileanu et al., *Text-to-Speech Synthesis for Romanian Language: Present and Future Trends*, http://www.racai.ro/books/awde/burileanu.html
[10] O. Buza, G. Toderean, "A Romanian Syllable-Based Text-to-Speech Synthesis", *Proceedings of the 6th WSEAS Internat. Conf. on Artificial Intelligence, Knowledge Engineering and Data Bases* (AIKED '07), Corfu Island, Greece, 16-19 February, 2007, CD
[11] O. Buza, G. Toderean, "About Construction of a Syllable-Based TTS System", *WSEAS Transactions on Communications*, Issue 5, Volume 6, May 2007, ISSN 1109-2742 , 2007
[12] O. Buza, G. Toderean, A. Nica, Zs. Bodo, "Original Method for Romanian Text-to-Speech Synthesis Based on Syllable Concatenation", *Advances in Spoken Language Technology*, coordinated by Corneliu Burileanu and Horia-Nicolai Teodorescu, ed. by The Publishing House of the Romanian Academy, composed of the Proc. of the 4th Conference on Speech Technology and Human Computer Dialogue "SpeD 2007", organized by the Romanian Academy, the University "Politehnica" of Bucharest, and the Technical University of Iasi, Iasi, Romania, May 10-12, 2007, p. 109-118