BUILDING A TEXT TO SPEECH SYSTEM FOR ROMANIAN THROUGH CONCATENATION

Ovidiu BUZA[*], Gavril TODEREAN[*], Joszef DOMOKOS[**], Arpad Zsolt BODO*

[*] Technical University of Cluj-Napoca
[**] Sapientia University of Târgu-Mureş

Corresponding author:

Ovidiu BUZA, Baritiu 26-28, Cluj-Napoca, Ovidiu.Buza@com.utcluj.ro

We present in this article our experience in building a text-to-speech system for Romanian through concatenation method. Main stages of this work were following: voice signal analysing and segmentation; building the vocal database; text analysing: pre-processing, unit detection, prosody retrieval; unit matching; unit concatenation and speech synthesis. In our approach we consider word syllables as basic units and stress indicating intrasegmental prosody. A special characteristic of current approach is rule-based processing of both speech signal analyse and text analyse stages.

Key words: text-to-speech, syllable approach, rule-based processing.

1. INTRODUCTION

In the last decades many methods have been developed for generating acoustical parameters requested for a high quality voice synthesis. Researches proved that among methods with best results are those methods which store the real acoustic waveform uttered by a human speaker. These methods achieve voice synthesis through concatenation of acoustic units, so they are called concatenation methods ([9],[14]).

The authors have worked on this line of attaining a voice synthesis complying with quality parameters of natural, human speech. Our researches led into projecting a voice synthesis method specifically adapted to Romanian language, and also into a working approach for constructing an automated speech synthesis system.

Using syllables as basic units, the projected method is integrated into high quality methods category, based on concatenation. We propose here an original approach based on rules that apply in the most important stages of projecting a speech synthesis system :

a) construction of the vocal database, by extracting acoustic units from speech using property rules,
b) text processing stage, by extracting linguistic units using phonetic and lexical rules.

The main stages in building our LIGHTVOX text-to-speech system were (fig.1) :
- voice signal analysis
- speech segmentation
- vocal database construction
- text analysis : pre-processing, unit separation and intrasegmental prosody detection
- unit matching
- unit concatenation and synthesis.

In the voice signal analysis stage, we have extracted main parameters of pre-recorded speech, working in time domain of analysis. These parameters, such as : amplitude, energy, zero-crosses, were used in the second stage for speech segmentation. In this second stage, we have designed an algorithm for automated speech segmenting in ten different classes of regions, that we have put into correspondence with main phonetic categories of Romanian language. After phonetic segmentation of speech signal, in the next stage we have used a semi-automated algorithm for detecting and storing waveform syllables from regular and special uttered words into vocal database.

In text processing stage, special phonetic rules have been developed for text pre-processing, syllables and intrasegmental prosody (i.e. stress) detecting. Next, unit matching was done by selecting acoustic units from vocal database according to the linguistic units detected from the input text. And finally, acoustic units are concatenated to form the output speech signal, that is synthesized by the mean of a digital audio card.

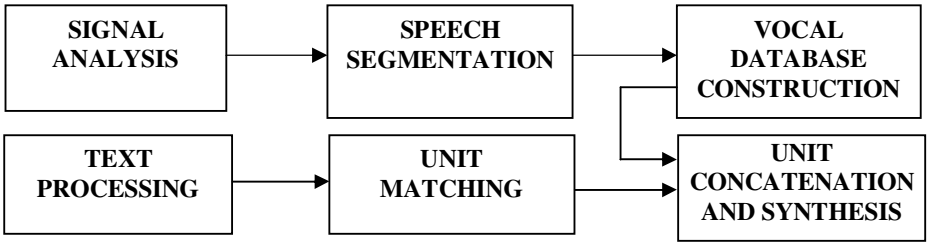

Figure 1. Main stages in building LIGHTVOX text-to-speech system

2. VOICE SIGNAL ANALYSIS

Voice signal analysis is the first pre-requisite stage in speech synthesis. Voice signal analysis means the detection of characteristics and signal parameters from speech samples recorded by the human speaker. The analysis can be done in time or frequency domain. Time domain analysis, as our approach is, leads to the detection of signal characteristics directly from waveform samples.

We have extracted following parameters: maximum and median amplitude, signal energy, number of zero-crosses and fundamental frequency.

Signal Amplitude gives information about presence or absence of speech, about voiced and unvoiced features of the signal on analyzed segment. In the case of a voiced segment of speech, as a vowel utterance, the amplitude is higher, beside the case of an unvoiced speech segment, where amplitude is lower.

Mean amplitude for N samples has the following form ([7]):

$$M(n) = \frac{1}{N} \sum_{m=-\infty}^{\infty} | x(m) | w(n-m) \quad , \qquad (1)$$

where: $x(m)$ represents the current sample of speech signal, and
$w(n-m)$ is the considered windowing function.

Signal Energy is used for getting the characteristics of transported power of speech signal. For a null-mean signal, short term mean energy is defined as [8]:

$$E(n) = \frac{1}{N} \sum_{m=-\infty}^{\infty} [x(n) \cdot w(n-m)]^2 \qquad (2)$$

Voiced segments (like vowels) have a higher mean energy, while the unvoiced segments (like fricative consonants) have a lower mean energy. For the majority speech segments, energy is concentrated in 300-3000 Hz band.

Number of zero-crosses is used for determining frequency characteristics inside a segment. Number of zero-crosses is calculated as follows ([7]):

$$NTZ = \frac{\sum_{n=0}^{N-1} [1 - \mathrm{sgn}(s(n+1)T) \cdot \mathrm{sgn}(s(nT))]}{2} \quad , \qquad (3)$$

where sgn(n) is the sign function:

$$\text{sgn}(n) = \begin{cases} +1, n \geq 0 \\ -1, n < 0 \end{cases} . \qquad (4)$$

Number of zero-crosses is a characteristic used in determining voiced/unvoiced feature of speech segments. Inside voiced segments number of zero-crosses is lower, while inside unvoiced segments this parameter has much higher values.

Fundamental Frequency is an important parameter used in voice synthesis that corresponds with the signal periodicity. This parameter must be computed on short segments of time, because of speech signal variability. Fundamental frequency is calculated only for voiced segments where the signal is almost periodical, while unvoiced segments are nonperiodical signals and they have not fundamental frequency.

For computing this parameter we have developed a time-domain method based on local maximum and minimum values of signal amplitude, as one can find in [9].

3. SPEECH SEGMENTATION

Finding an optimal approach for speech signal segmentation is an imperative in building acoustic database of a voice synthesis system. This section presents a segmentation method that have been designed and implemented by the authors, method which is capable to detect SUV (*Silence- Unvoiced- Voiced*) components of speech signal, to divide these components in regions with specific characteristics, and to associate regions with a known phonetic input sequence (figure 2):

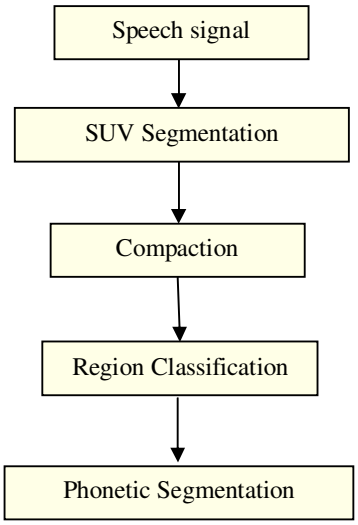

Figure 2. Speech signal segmentation

3.1. SUV Segmentation

Our segmentation method uses time domain analysis of speech signal. After low-pass filtering of signal, zero-crossing waveform points (Z_i) are detected. Then minimum (m_i) and maximum (M_i) values between two adjacent zero points are computed.

Separation between silence and speech segments is realised by using a threshold value T_s on signal amplitude. In ***silence*** segments, all m_i and M_i points must be lower than this threshold:

$$\begin{cases} |M_i| < T_s \\ |m_i| < T_s \end{cases}, i = s\ldots\ s+n, \tag{5}$$

where s is the segment sample index and n is the number of samples in that segment.

For speech segments, distance D_i between two adjacent zero points is computed.
Decision of ***voiced*** segment is assumed if distance is greater than a threshold distance V:

$$D_i > V\ , \quad i = s,\ldots,\ s+n \tag{6}$$

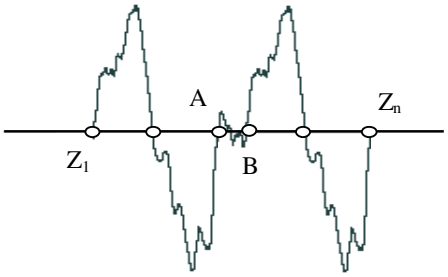

Figure 3. A voiced segment of speech

For the zero points between A and B from figure 3 to be included in the voiced segment, a look-ahead technique has been applied. A number of maximum N_k zero points between Z_i and Z_{i+k} can be inserted in voiced region if $D_{i-1} > V$ and $D_{i+k} > V$:

$$\begin{cases} D_j < V & , j = i..k; \quad k <= N_k \\ D_{i-1} > V & \\ D_{i+k} > V & \end{cases}, \quad i = s,\ldots,\ s+n \tag{7}$$

A segment is assumed ***unvoiced*** if distance between two adjacent zeros is smaller than a threshold U:

$$D_i < U\ , \qquad i = s,\ldots,\ s+n \tag{8}$$

Transient segments are also defined and they consist of regions for which conditions (6), (7) and (8) are not accomplished. In these relations, V and U thresholds have been chosen according to statistical median frequency for vowels and fricative consonants.

3.2. Compacting regions

After first appliance of above algorithm, a large set of regions will be created. Since voiced regions are well determined, the unvoiced are broken by intercalated silence regions. This situation appears because unvoiced consonants have low amplitude so they can break in many silence/unvoiced subregions.

Transient segments can also appear inside the unvoiced segment because of signal bouncing above zero line.

Figure 4 shows such an example, in which numbered regions are unvoiced, simple-line and unnumbered are silence regions, and double-line are transient regions.

All these regions will be packed together in the second pass of the algorithm, so the result will be a single unvoiced region – as one can see in figure no. 5.

After segmentation, voiced and unvoiced segments are coupled according to the syllable chain that is used in vocal database construction process. Appropriate acoustic units will be detected, labeled and stored in vocal database.

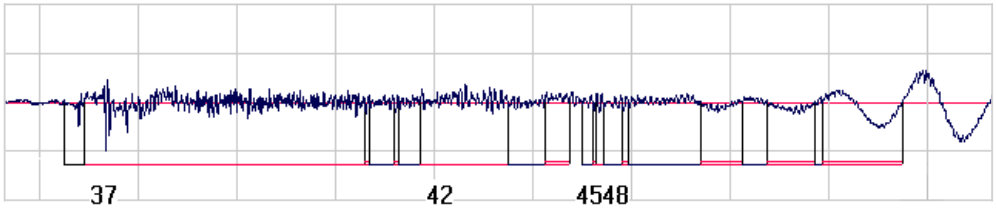

Figure 4. Determining regions for an unvoiced segment of speech

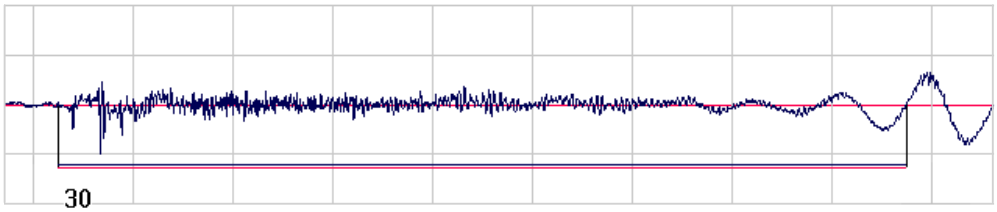

Figure 5. Compacting regions of above segment

3.3 Regions classification

The SUV segmentation process presented above divides the signal in four basic categories: ***S****ilence,* ***V****oiced,* ***U****nvoiced* and ***T****ransition.* Each category will be further classified in distinct types of regions, totally 10 classes: silence, unvoiced-silence, voiced, voiced glide, voiced plosive, voiced jump, transition, irregular (rugged), high density and unvoiced consonant (figure 6). The aim of this classification is to associate Romanian phonemes with signal regions having some particular traits.

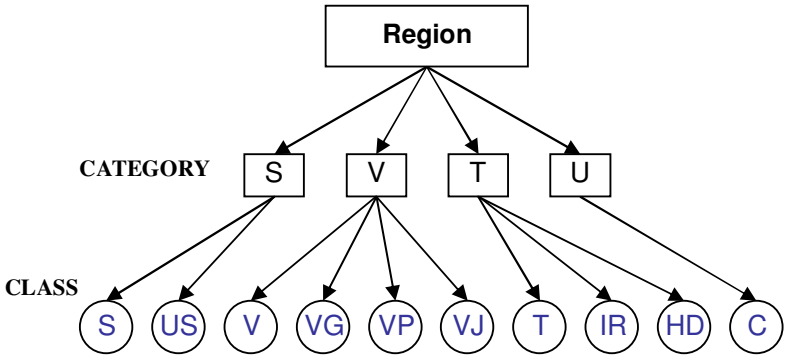

Figure 6. The four categories and ten classes of regions

These ten classes of regions are shortly presented as follows:

1. Silence Region (S)

Represents a region without speech, where signal amplitude is very low.

2. Unvoiced Silence (US)

This region is a combination of silence ***S*** and unvoiced consonant ***C*** region**.** Detecting of this region as separate class was necessary because of fricative consonants that can be uttered at low amplitude, and so they could be found in these ***US*** regions.

3. Voiced Region (V)

The voiced region contains all vowels from Romanian: /A/, /E/, /I/, /O/, /U/, /Ă/, /Î/, glide /L/, nasals /M/, /N/, and some voiced plosive consonants as /P/, /B/, /D/.

4. Voiced Glide (VG)

This is a region corresponding to a voiced discontinuity and is associated with a minimum of energy. This situation may occur when a glide consonant like /R/ splits a sequence of vowels.

5. Voiced Plosive (VP)

Also a region of voiced discontinuity corresponding to intermediate frequencies associated with plosive consonants like /C/ or /G/, occurring when these consonants are splitting a sequence of vowels.

6. Voiced Jump (VJ)

Is a region similar with voiced region ***V***, but it has no periodicity. It is due to the balance of vocal signal only above or underlying median zero line. This region has no vowel or other phoneme correspondence in speech signal, but a transition or co-articulation.

7. Irregular Region (IR)

This is a region in which one can find plosive consonants like /C/, /G/ or /P/. Usually it comes after a silence region, it has a short duration and a frequency band intermediate between vowels and fricative consonants.

8. High Density transition (HD)

Is a transition region with high frequency values, which could indicate emergency of fricative consonants. Signal is not integrated in ***C*** or ***US*** classes because of positive or negative balance relative to median zero value.

9. Transition (T)

Is an intermediate region between voiced and unvoiced and which has no the characteristics of ***IR*** or ***HD*** classes.

10. Unvoiced Consonant (C)

For Romanian language, this class contains fricative consonants /S/, /Ş/, /Ţ/, /F/, /Z/, /J/, /H/, and non-fricatives /Ce/, /Ci/, /Ge/, /Gi/ .

For detection of all these classes, median amplitude *MA(n)*, number of zero-cross points *NTZ*, signal energy *E(n)* have been used as section 2 describes in (1)-(4), and also short-term Fourier coefficients for detection of ***VP*** and ***HD*** special cases.

3.4. Phonetic segmentation

Phonetic segmentation is the process of associating phonetic symbols with the speech signal. This process is very usefull when we want to develop an acoustic database from a large speech corpus. Phonetic segmentation gives the capability of detecting and separating phonetic units from speech, units that will be used in achieving the output acoustic chain sequence through concatenation.

State of the art on this field exposes some different methods for automated or semi-automated phonetic segmentation of speech signal: iterative methods with training stages (as HMM segmentation or region frontiers refinement), methods based on association rules, statistical methods (like segmentation based on generalized likelihoods GLR), a.o.

Phonetic sgmentation method proposed by the authors is a method based on association rules, which realises a correspondence between phonetic groups taken from an input stream and distinct types of regions detected from the speech signal. Segmentation algorithm parses the input text and tries to find the best match for each phonetic group with one or more regions of speech signal.

Input text is first written into a special phonetic transcription, using a simple look-after table, which includes phoneme and word transitions. Transcripted text is splitted into a sequence of phonetic groups. Special association rules will establish a correspondence with specific regions detected from speech signal.

For associating phonetic groups with sequences of regions, as further in our approach, we have used LEX parser generator. We have written a set of association rules, each rule specifying a phonetic pattern for associating a particular group with a sequence of regions, and also specifying a condition to be verified in order to make that association. Each condition outlines: type of region, minimal and maximal duration, type of association: unique region or sequence of regions.

Figure no. 7 presents association between a phonetic group $G_i = \{F_1^i, F_2^i, ..., F_k^i\}$, where F_k^i are phonetic symbols, and a sequence of regions $SR_N = \{REG_1^i, REG_2^i, ..., REG_N^i\}$, by the meaning of a correspondence rule $\boldsymbol{R_i}$ **:** $G_i <> COND\_REG_i$.

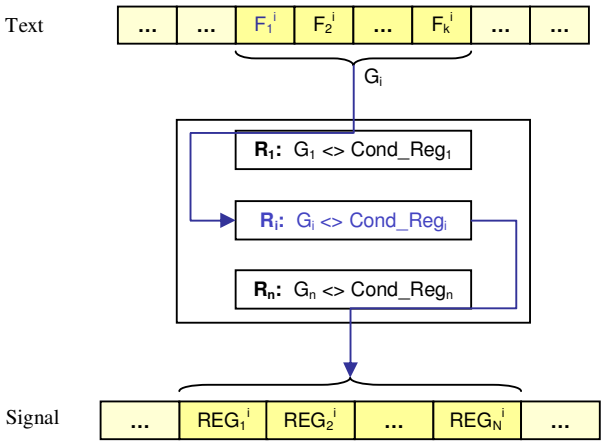

Figure 7. Association between phonetic groups and signal regions

Here are two samples of rules for associating with regions:
(a) a generic fricative consonant and
(b) a specific group of consonants:

```
{CONS}  { /* FRICATIVE CONSONANT */
            CheckRule(i);  // check processing rule for current group for
                           // going only forward
            SetLen(L_CONS1,L_CONS2); // set minimum and maximum duration
            CheckRegion(R_CONS);   // check next region to be an unvoiced
                                   // consonant
            TestReject();   //if  above  conditions  are  not  complied,
                            //rejects the rule and go to the next matching
            }

TR    {    /* GROUP OF TWO CONSONANTS: /TR/ */
            CheckRule(j); // check processing rule
            SetLen(L_TR1,L_TR2); // set minimum and maximum duration
            CheckSumReg(R_ANY && !R_VOC); // check a sequence of regions
                                          // of any type but not voiced
            TestReject();    // if above conditions are not complied,
                            //rejects the rule and go to the next matching
            }
```

In figure 8 one can see the result of applying our method of association phonemes-regions on a sample male utterance:

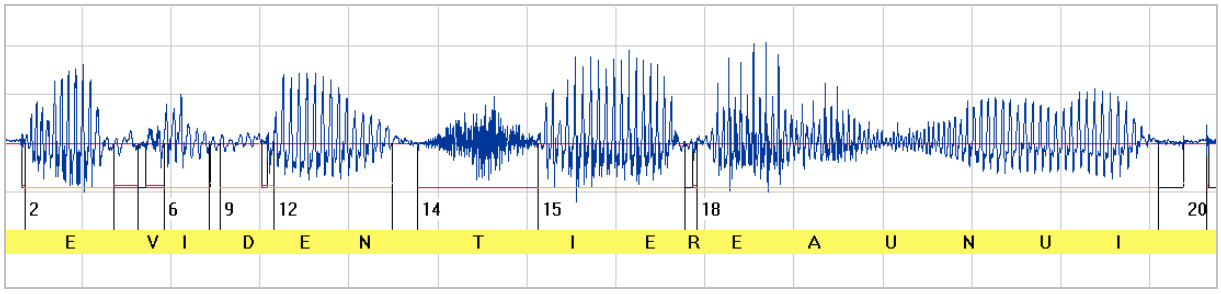

Figure 8. Phonetic segmentation for the expression: <*EVIDENŢIEREA UNUI*>

4. BUILDING THE VOCAL DATABASE

Phonetic segmentation method described in previous section has been designed for segmentation and labelling of speech corpora, having as main objective the construction of vocal database of our speech synthesis system. In our approach, realisation of vocal database implies separation of acoustic segments that correspond with phonetic syllables of Romanian language and storing these segments into a hierarchical structure. Vocal database includes in this moment only a subset of Romanian language syllables. We have not considerred in this implementation diphones.

The speech corpus used for extracting acoustic units was built from common Romanian sentences, from separate words containing syllables, and also from artificial words constructed for the emphasis of one specific syllable. After recording, speech signal was normalized in pitch and amplitude. Then phonetic segmentation was applied and acoustic syllables were stored in database.

The vocal database has a tree data structure; each node in the tree corresponds with a syllable characteristic, and a leaf represents appropriate syllable.

Units are stored in database following this classification (fig.9):
- after length of syllables : we have two, three or four characters syllables (denoted S2, S3 and S4) and also singular phonemes (S1);
- after position inside the word: initial or median (M) and final syllables (F);
- after accentuation: stressed or accentuated (A) or normal (N) syllables.

S2 syllables, that are two-character syllables, have following general form:
- {CV} (C=consonant, V=vowel), for example: ‚ba', ‚be', ‚co', ‚cu', etc, but we have also recorded syllable forms like:
- {VC}, as ‚ar', ‚es', etc., forms that usually appear at the beginning of Romanian words.

S3 syllables, composed from three phonemes, can be of following types:
- {CCV} , for example: ‚bra', ‚cre', ‚tri', ‚ghe';
- {CVC} , like: ‚mar', ‚ver';
- {CVV} , for example: ‚cea', ‚cei', ‚soa'.

S4 four-character syllables have different forms from {CCVV} , {CVCV} to {CVVV}.

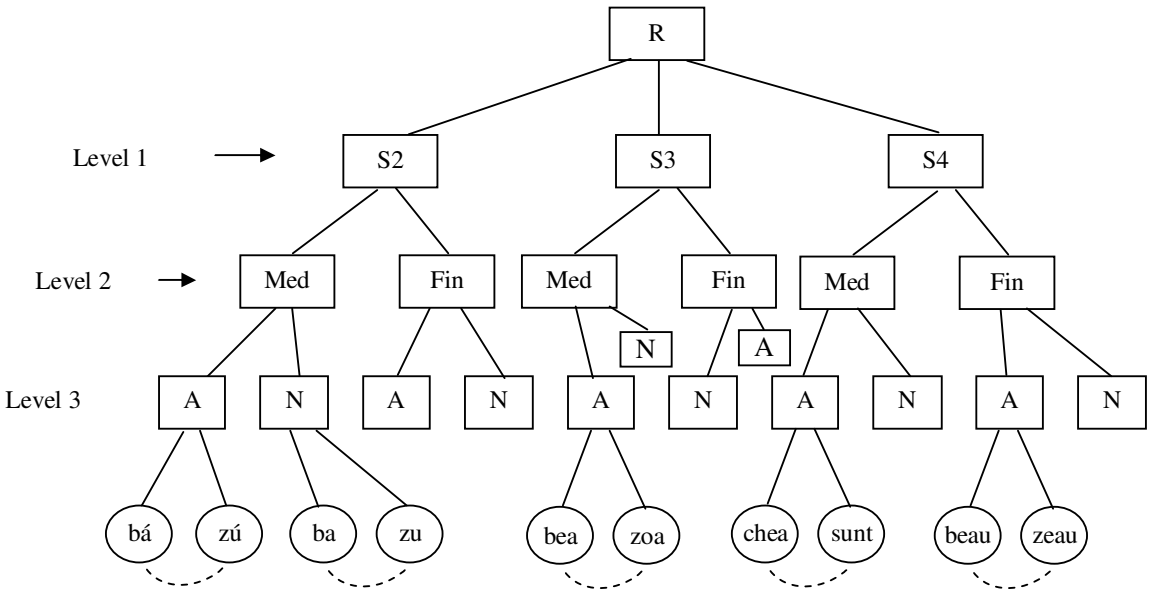

Figure 9. Database hierarchical structure

5. 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 ([3]).

The principle used in detecting linguistic units is illustrated in figure no. 10. Here we can see the structure of text analyzer that corresponds to four modules designed for detection of units, prosody information and unit processing.

These modules are:

- a lexical analysis module for detection of basic units;
- a phonetic analysis module for generating prosody information;
- a high level analysis module for detection of high-level units;
- the processing shell for unit processing.

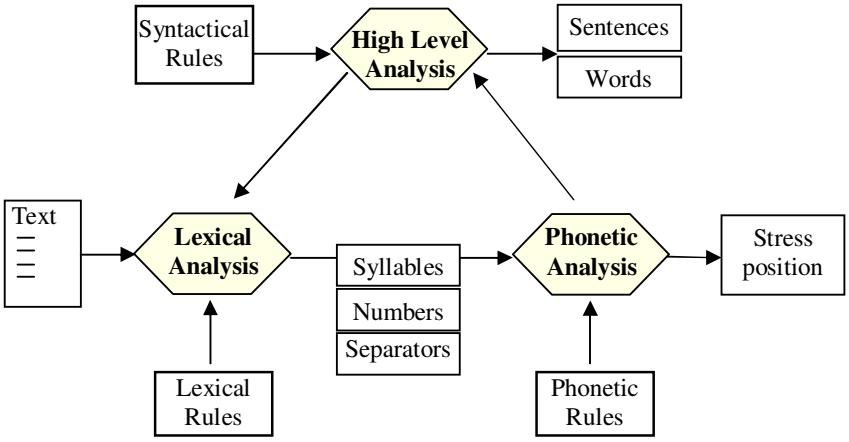

Figure 10. Text analysis for syllable detection

Lexical analyzer 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 [9] - [13]), 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 analyzer gets the syllables between two breaking characters and detects stress position, i.e. the accentuated syllable from corresponding word.

Then, high-level analyzer takes the syllables, special characters and numbers provided by the lexical analyzer, 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 these structures, synthesis module will construct the acoustic waves and will synthesize the text.

5.1. 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.

Hereby, input text is interpreted as a character string. At the beginning, current character is classified in following categories: digit, alphanumeric character, and special 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 subclassification of units (integer or real numbers, type of separators: word or sentence separator, etc).

A syllable-detection rule may have following general forms:

```
{ROOT_PATTERN}    { Proceed_Syllable;}                          (F1)

{ROOT_PATTERN}/{PATTERN}    { Proceed_Syllable;}                (F2)

{PATTERN}{TERMINATION}/{SEP}    { Proceed_Syllable;}            (F3)
```

Rule (F1) is applied for diphthongs like /OA/ or /IU/ that always occur in same syllable inside the root of a word, regardless of subsequent (right) context. Rule (F2) applies for middle-word syllables, since rule (F3) applies for ending-word syllables (having a right context of a word separator).

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

The basic set shows the general decomposition rules for Romanian.

First rule is that a syllable consists of a sequence of consonants followed by a vowel:

```
syllable ={CONS}*{VOC}                                          (R1)
```

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

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

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 :

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

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 180 rules. Rules are grouped in subsets that refer to resembling character sequences. All these rules were completely explained in [9].

5.2. Syllable accentuation

The principle for determining syllable accentuation resembles with that of lexical analyzer for detecting syllables already exposed. After the text parser returns from input stream current word consisting of phonemes F_1, F_2, …, F_k and delimited by a separator, phonetic analyzer reads this word and detects syllable accentuation based on phonetic rules. Rules have been also written in BNF form and set into LEX input.

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 next to last position.

The rules set for determining accentuation consists of:

a) One general rule meaning S_{n-1} syllable is stressed:

```
{LIT}+/{SEP}     { return(SN-1);}
```
(G1)

and

b) A consistent set of exceptions, organized in classes of words having the same termination. Each rule from exceptions set presents following form:

```
{PATTERN}{TERMINATION}/{SEP}    { return(SN-x);}
```
(E1)

where *x* can be one of 0, 1, 2, 3.

At this time, the exception set is made up by more then 250 rules. All these rules were presented and completely explained in [9].

6. UNIT MATCHING, CONCATENATION AND SYNTHESIS

Matching process is done according to the three–layer classification of units: number of characters in the syllable, accentuation and the place of syllable inside the word.

If one syllable is not found in vocal database, this will be constructed from other syllables and separate phonemes that are also recorded. Following situations may appear:

(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 at all, so it will be constructed from separate phonemes.

After matching, units are simply concatenated to result the acoustic chain that will be synthesized. In this stage of development, our system works with intrasegmental prosody i.e. accentuation inside words, and doesn't support sentence-level prosody like intonation. The rhythm of speech can be adjusted by intercalating different periods of silence between syllables, words and sentences.

7. IMPLEMENTATION

The purpose of our work was to build a speech synthesis system based on concatenation of syllables. The system includes a syllable database in which we have recorded near 400 two-character syllables, 150 most frequent three-character syllables and 50 four-character syllables. Syllables that are not included in database are synthesized from existing syllables and separate phonemes that are also recorded.

The speech synthesis system first invokes text analyzer for syllable detection, then phonetic analyzer for determining the accentuation. Appropriate units (stressed or unstressed) are matched from vocal database, and speech synthesis is accomplished by syllable concatenation.

8. CONCLUSIONS AND RESULTS

We have presented in this article a complete method for building a syllable-based text-to-speech system. First, speech signal was segmented in basic categories and ten different classes of regions. Then, a rule-based phonetic segmentation method was invoked onto a speech corpus in order to associate input phonemes with regions. We have used this phonetic segmentation method to separate phonetic units from speech corpora and create the vocal database.

Special efforts have been done to accomplish the text processing stage. Here we have designed two sets of rules: one set of rules for detecting word syllables and a second set for determining the accentuation inside each word. Even these sets are not complete, they cover yet a good majority of cases. The lexical analyzer is based on rules that assure more than 98% correct syllables detection, since accentuation analyzer provides about 93% correct detection rate (computed on near 50000 words collection consisting of various Romanian texts from literature, religion, science and technical fields).

The advantages of detecting syllables through a rules-driven analyzer are: separation between syllables detection and system code, facile 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, the results are encouraging, and after a post-recording stage of syllable normalization we have obtained a good, near-natural quality of speech synthesis. Even diphones have not been considerred in our method, the speech outcome is not affected. For the future implementations, we have in mind the completion of syllable and accentuation rules sets and also the completion of syllables database according to a self-generated statistics of Romanian syllables, aspects that will improve the system performance.

REFERENCES

1. BURILEANU, D., 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. BURILEANU, C., et al., *Text-to-Speech Synthesis for Romanian Language: Present and Future Trends*, in the volume "Recent Advances in Romanian Language Technology" (D. Tufiş, P. Andersen – Eds.), Publishing House of the Romanian Academy, Bucharest, pp. 189-206, 1997.
3. CIOMPEC, G., et al., *Limba română contemporană. Fonetică, fonologie, morfologie*, Editura Didactică şi Pedagogică, Bucharest, 1985.
4. FREE SOFTWARE FOUNDATION*, Flex - a scanner generator,* http://www.gnu.org /software/flex/manual, October 2005.
5. HUNT, A., BLACK, A., *Unit selection in a concatenative speech synthesis system using a large speech database*, IEEE International Conference on Acoustics, Speech and Signal Processing ICASSP '96 Proceedings, Atlanta, GA, pp. 373–376, May 1996.
6. LEWIS, E., TATHAM, M., *Word And Syllable Concatenation In Text-To-Speech Synthesis***,** Sixth European Conference on Speech Communications and Technology, pages 615—618, ESCA, September 1999.
7. PICONE, J.W., Signal modeling techniques in speech recognition, Proceedings IEEE vol. 81 sept. 1993 pp. 1215-1246.
8. LUPU E., POP P., *Prelucrarea numerică a semnalului vocal*, vol.1, Ed. Risoprint, 2004.
9. BUZA, O., *Vocal interractive systems*, doctoral paper, Electronics and Telecommunications Faculty, Technical University of Cluj-Napoca, 2005
10. BUZA, O., TODEREAN, G., *Syllable detection for Romanian text-to-speech synthesis*, Sixth International Conference on Communications COMM'06 Bucharest, June 2006, pp. 135-138.
11. BUZA, O., TODEREAN, G., *A Romanian Syllable-Based Text-to-Speech Synthesis*, Proc. of the 6th WSEAS Internat. Conf. on Artificial Intelligence, Knowledge Engineering and Data Bases (AIKED '07), CD Proceedings, Corfu Island, Greece, 16-19 February, 2007,
12. BUZA, O., TODEREAN, G., *About Construction of a Syllable-Based TTS System*, WSEAS TRANSACTIONS on COMMUNICATIONS, Issue 5, Volume 6, May 2007, ISSN 1109-2742 , 2007
13. BUZA, O., TODEREAN, G., NICA, A., BODO, Z., *Original Method for Romanian Text-to-Speech Synthesis Based on Syllable Concatenation*, published in the volume "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, pp. 109-118, May 10-12, 2007
14. BUZA, O., TODEREAN, G., *Metode de Sinteză din Text a Vorbirii pentru Limba Română*, The Second International Conference "Telecommunications, Electronics and Computer Science" ICTEI 2008, Chişinău, Republica Moldova, pp. 209-214, 15-18 of May 2008.