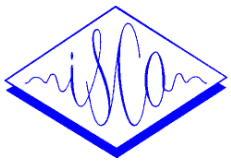

The ESCA Workshop
on Speech Synthesis
Autrans, France
September 25-28, 1990

CONCEPT DESCRIPTION FOR SYNTHETIC SPEECH OUTPUT SYSTEM

Yoichi Yamashita*, Naoki Mizutani and Riichiro Mizoguchi***

* The Institute of Scientific and Industrial Research, Osaka University,
8-1, Mihogaoka, Ibaraki-city, Osaka, 567 Japan.
** SHARP corporation,
2613-1, Ichinomoto, Tenri, Japan.

ABSTRACT

This paper describes a concept description scheme for speech synthesis. It is input to the synthetic speech output interface connected to various performance systems, and used for direct derivation of prosodic parameters. The concept description is composed of atomic symbols, templates and operators represented in terms of appropriate abstract level of constructs and makes it easy to generate both sentences and prosodic parameters. There are two built-in mechanisms in the templates for directly controlling the prosodic parameters. The first one is the pause marker which is generated along with words in the sentence generation. The pause marker is used to insert pauses and to locate boundaries of phrase component of pitch. The second one is the Prosody Modification Functions (PMF) embedded in the custom templates. PMF controls the the prosodic parameters for the prepared sentence pattern.

1. INTRODUCTION

Speech synthesis from concept descriptions is a very important technique for the synthetic speech output system and has several advantages over speech synthesis from text [Young,1979]. One of the goals of speech synthesis research is to construct the universal Synthetic Speech Interface (SSI) as the frontend of various Intelligent Computer Systems (ICS). To output a speech message to the user, the ICS has to send a sort of description of the message to SSI. The description can be written in several forms: texts, descriptions using Conceptual Dependency theory [Schanck,1975], and so on. The authors proposed a description scheme like a parsing tree based on the case structure, which is referred to as the concept description. The concept description is designed as the input of the SSI from the ICS. In general, text is not a best description because it is not easy to extract the information necessary for the prosody control from it. The proposed concept description can make it easy to control prosodic patterns and avoid the complexity of the sentence generation which the abstract descriptions require in SSI.

2. CONCEPT DESCRIPTION

The SSI receives the concept description as the output request from the ICS. The SSI generates a sentence and prosodic parameters and the synthesizer produces the speech sound. The concept description is composed of seven elements which are shown in the following:

(1) Atomic symbol: Verbs, nouns and adjectives which appear in a generated sentence are described by themselves as the atomic symbols.

(2) Modification template: The modification template, $modify(A,B,OPs), describes the modification relation between the two arguments, A and B, which are constructs of sentences such as nouns, adjectives and phrases. This template contains different patterns dependent on the first argument, A, as shown in Fig.1. In sentence generation, the template generates words using the output word list prepared for each pattern which is denoted in brackets in Fig.1. The symbol 'p' is a pause marker which is generated along with words and used later in prosody control. The third

```
$modify( A, B, OPs ) : / the final word of A is a noun / [ A, の, p, B ].
                                                    ( B of A )
                       / the final word of A is an inflective word / [ A*, B ].
                       / otherwise / [ A, B ].    ( A*: modification of A )
```

Fig.1 Modification Template.

```
$actor(A):       [ A, が, p ].
$object(A):      [ A, を, p ].
$subject(A):     [ A, に, p ].
$instrument(A): [ A, で, p ].  ( with A )
$means(A):       [ A, に, よって, p ]. ( by A )
$from(A):        [ A, から, p ].  ( from A )
$to(A):          [ A, まで, p ].  ( to A )
$direction(A):   [ A, へ, p ].  ( to/for A )
$place(A):       [ A, で, p ].  ( in/at A )
$time(A):        [ A, p ].
$range(A):       [ A, まで, p ].  ( to A )
```

Fig.2 Examples of Case Templates.

```
$progress:     ～ている  ( ～ing )
$perfect:      ～てしまう  ( have <pp.> )
$past:         ～た  ( <past> )
$future:       ～だろう  ( will ～ )
$negative:     ～ない  ( not ～ )
$possible:     ～できる  ( can ～ )
$imperative:   ～なさい
$interrogative: ～か
$persuade:     ～う  ( let's ～ )
$passive:      ～れる, られる  ( be <pp.> )
```

Fig.3 Examples of Mood Templates.

```
$conversion(A): [ それでは, p2, A ].  ( Well, A )
$addition(A):   [ そして, p2, A ].  ( And, A )
$pos_cond(A):   [ 従って, p2, A ].  ( Therefore, A )
$neg_cond(A):   [ しかし, p2, A ].  ( But, A )
```

Fig.4 Examples of Conjunctive Templates.

```
$same( A, B, OPs ) : [ A, と, 同じ, p, B ], $mod_acc(1,+,15).
                                    ( the same B as A )
$enum( A, B, C, OPs ) : [ A, や, p, B, など, の, p, C ].
                                    ( C such as A and B )
$reason( A, B, OPs ) : [ A, は, p2, B, から, だ ], $mod_acc(4,+,15), $mod_bpit(4,+,10).
                                    ( the reason of A is B )
$because( A, B, OPs ) : [ A, から, p2, B ], $mod_bpit(4,190).
                                    ( Because A, B )
$why( A, OPs ) : [ なぜ, p2, A, の, です, か ], $mod_dur(1,+,10), $mod_fpit(3,190).
                                    ( why A )
```

Fig.5 Examples of Custom Templates.

```
s1( 始める([$object(lesson1),$persuade]) ).
    (begin)
lesson1( $modify( 地理, 勉強, []) ).
                  (geography) (study)
s2( 生育する([$actor(米),$place(place1),$interrogative,$p_interrogative]) ).
    (glow)          (rice)
place1( $modify( 寒い, 所, [$p_prominence(寒い)]) ).
                  (cold) (place)          (cold)
s3( $because(soviet1,生育する([$actor(米),$negative])) ).
                        (glow)          (rice)
soviet1( $is( ソ連, $modify( 寒い, 国, []) ).
              (the Soviets)    (cold) (country)
s4( $why(生育する([$actor(米), $place(イラン)]),[$p_interrogative]) ).
          (glow)          (rice)        (Iran)
```

Fig.6 Examples of Concept Description.

argument, OPs, is a list of prosody operators mentioned later.

(3) Case template: The case templates are prepared to describe the case information of verbs and appear as the arguments of verbs in the concept description. These templates produce function words in the sentence generation, as shown in Fig.2, because the function words express the case in Japanese sentences.

(4) Mood operator: The mood operators are prepared to describe the tense, aspect and modality information of verbs. They also appear as the arguments of verbs in the concept description. The mood information is expressed by the conjugation of the verb and the following function words in Japanese. The mood operator changes the verb into the appropriate form according to the pre-stored pattern. Fig.3 shows some examples of the mood operators.

(5) Conjunction template: The concept descriptions are processed sentence by sentence. The conjunction templates describe the relation to the preceding sentence. This template requires a sentence as its argument, and generates the conjunction before the sentence in the sentence generation, as shown in Fig.4.

(6) Custom template: The custom templates describe the frequently used expressions and the difficult expressions to describe based on the case. In the custom template, PMF can locally change the prosodic parameters. Fig.5 shows some examples of the custom templates. $mod_acc in $reason template is an example of PMF. Thus, PMFs facilitate the control of prosody for these expressions. It is discussed in the next section in detail.

(7) Prosody operator: The prosody operators indicate the prosodic features and are used as arguments of verbs or templates mentioned above. Three kinds of prosody operators are defined as mentioned in 4-3.

Examples of the concept description are shown in Fig.6.

3. SENTENCE GENERATION

Sentences are generated according to template matching and case grammar. The case information for each verb is output in a pre-determined order. Fig.7 shows sentences generated from the examples in Fig.6. The template has the list of output words which includes pause markers, 'p' and 'p2'. Pause markers are generated in sentences as shown in Fig.6, and are utilized to both insert pauses and boundaries of prosodic phrases in the prosody control.

4. PROSODY CONTROL

4-1. Pause Insertion

Pause insertion is necessary to the naturality of synthetic speech because the human utterance duration in a breath is limited. Two kinds of pauses are used: PS2 is a long pause (300ms) and PS1 is a short one(100ms). Pauses are inserted according to the pause markers embedded in generated sentences. The 'p2' indicates the insertion of PS2 and the 'p' indicates the potential insertion of PS1. The pause marker 'p2' is placed at the word boundary in the output word list of

s1: 地理の p 勉強を p 始めましょう
(Let's begin the study of the geography.)

s2: 米が p 寒い所で p 生育しますか
(Does the rice glow in the cold place ?)

s3: ソ連は p 寒い国ですから p2 米が p 生育しません
(The rice does not glow, because the Soviets is a cold country.)

s4: なぜ p2 米が p イランで p 生育するのですか
(Why does the rice glow in Iran ?)

Fig.7 Generated Senteces.

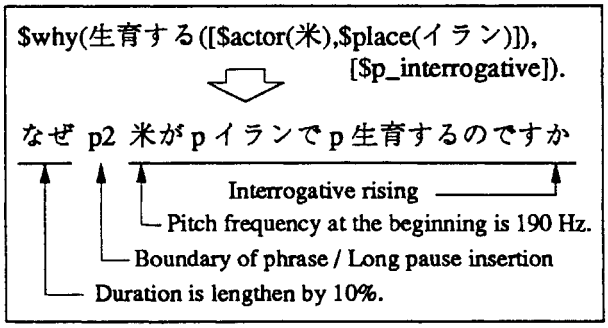

Fig.8 An Example of Prosody Control.

the templates, shown in Figs.4 and 5, where a long pause would always appear when the sentence is uttered by the human.

Some of the 'p's are replaced by PS1 taking account of number of the morae between the pauses, while every 'p2' is always replaced by PS2. If the number of morae in a one-breath phrase is over the threshold(=25), one of 'p' in the phrase is replaced by PS1 to divide the phrase into two one-breath phrases. In sentence generation, templates generate words and pause markers and results in the nest structure. A 'p' from the most inside template is selected, when more than one 'p' are found in the phrase to be divided. The replacement of pause markers is recursively carried out.

4-2. Boundary of Phrase Component of Pitch

The pitch contour is controlled based on the model of the addition of the lexical word accent pattern to the declination line, and the phrase component of pitch is represented by the declination line. The boundary of the phrase component is determined in the same way as the pause insertion except using the smaller mora threshold (=15). Thus, the pause always makes the boundary of the phrase component of pitch.

4-3. Prosody Operators

The prosody operators described in the concept description can adjust prosodic parameters. Three prosody operators are currently defined: $p_prominence(A), $p_interrogative and $p_speed(A). $p_prominence increases the accent component of pitch and prosodically emphasizes its argument A. $p_interrogative requires that the pitch rises at the end of the sentence. $p_speed(A) controls the utterance speed of its argument A, partially in the sentence.

4-4. Prosody Modification Function (PMF)

The Prosody Modification Functions (PMF) are used only in the custom template while the prosodic operators are described in the concept description. Three PMFs, $mod_dur, $mod_bpit and $mod_acc, are prepared to control the duration, the pitch frequency at the beginning point of phrase component and the accent prominence component, respectively.

4-5. An Example of Prosody Control

Fig.8 shows an example of prosody control. Four modifications are carried out in this example. Both the duration for first word and the pitch frequency at the beginning of second phrase are derived from the PMFs in the custom template $why(A,OPs). The boundary of the phrase and the long pause insertion are determined by the use of the pause marker. And, the prosody operator, $_interrogative, in the concept description gives the pitch rising at the end of the sentence.

5. CONCLUSION

This paper proposed the concept description for the Synthetic Speech Interface (SSI) and a method of the direct control of the prosodic parameters for the synthetic speech using the concept description. The idea that prosody markers and PMFs control the prosody is useful for any language, though templates are dependent on the language. Our system of the synthetic speech output from the concept description has been implemented in C-Prolog, adopting a CAI system as the Intelligent Computer System.

REFERENCE

Schank R.C. (1975), "Conceptual information processing", 3, 22-82, North-Holland publishing company.

Young S.J. & Fallside F. (1979), "Speech synthesis from concept: A method for speech output from information systems", J.A.S.A., 66, 3, 685-695.