Message-to-Speech: high quality speech generation for messaging and dialogue systems

P. Spyns (1), **F. Deprez** (1), **L. Van Tichelen** (1), **B. Van Coile** (1,2)
(1) Lernout & Hauspie Speech Products, Sint Krispijnstraat 7, B-8900 Ieper, Belgium
tel.: 32-57-22.88.88, fax: 32-57-20.84.89
(2) E.L.I.S., University of Gent, Sint-Pietersnieuwstraat 41, B-9000 Gent, Belgium
tel.: 32-9-264.33.95, fax: 32-9-264.35.94
{Peter.Spyns,Filip.Deprez,Luc.VanTichelen,Bert.VanCoile}@lhs.be

Abstract

In this paper, we present a Message-to-Speech (MTS) system that offers the linguistic flexibility desired for spoken dialogue and message generating systems. The use of prosody transplantation and special purpose prosody models results in highly natural prosody for the synthesised speech.

1 Introduction

Many of the Natural Language Generation (NLG) systems that produce flexible output, i.e. sentences with variations on the syntactical and morphological levels, only aim at the production of written text and do not deal with spoken language. As a result, the important topic of generation of natural prosody is not touched upon (see e.g. (Elhadad, 1992; Reiter et al., 1995; Dalianis, 1996b; Somers et al., 1997)).

Message generating systems (e.g. announcement systems, phone banking and voice mail applications) often combine fixed pieces of pre-recorded speech to provide speech of a natural quality. In practical applications, the linguistic flexibility of the spoken messages is usually kept very limited because of the high costs of recording and storing the fixed pieces of speech.

The Message-to-Speech (MTS) system described below is specifically designed to generate high quality speech output with the flexibility desired for spoken dialogue and message generating systems. Such systems typically generate speech for a predefined set of messages that consist of fixed and variable parts. High flexibility may be required for the variable parts in the messages only.

Text-to-Speech (TTS) is an evident technique for providing speech output with nearly unlimited flexibility. As full flexibility is only needed for the variable parts in the messages, the MTS system can make use of special purpose prosody models for the actual set of messages of the application. These models can lead to a prosodic quality that is superior to the one generated by TTS systems, which apply general prosody models for unrestricted text (see also (Hovy, 1995, p.161)).

For the fixed parts of a message, the prosody transplantation technique (see section 2) is used to overrule prosody generated by general models, as is done by TTS, with specific prosody copied from natural speech. For the parts of a message where flexibility is needed, prosody is obtained by either a general model or by a model that is specifically developed for those parts. The MTS system thus combines transplanted prosody with prosody by model in order to achieve highly natural prosody for partly variable messages (Van Coile et al., 1995).

The key concepts of the MTS system are presented in section 3.1. The system consists of two main modules: a generation module and a prosodic integration module. The generation module (see section 3.2) is template driven (canned "text" interspersed with slots), and accounts for the flexibility, including the linguistic variation, of the messages. For a discussion of template driven systems see (Reiter, 1995; van Deemter et al., 1994; van Deemter and Odijk, 1997). The prosodic integration module (see section 3.3) takes care of the prosodic integration of the slot fillers with the rest of the template.

In section 4 the Message-to-Speech system is briefly discussed, and section 5 compares the system with related research. To conclude, an overview of current developments to further enhance the MTS system is presented in section 6.

2 Prosody Transplantation

The idea behind Prosody Transplantation is that of copying intonation and duration values from a recorded *donor* message (human speech) to the phonetic transcription of the same message. The *En-*

riched Phonetic Transcription (EPT) obtained in this manner can be fed to a TTS system whereby the normal linguistic and prosodic modules (based on general models) are by-passed (Phonetics-to-Speech — PTS). Only the segmental synthesis and the synthesiser modules are used.

```
# T[104] æ[74(0,98)] N[47] k[107(10,81)] j[14(0,106)]
u[44] f[93(0,91)] o[47(0,102)] r[29] j[68(0,98)(30,90)]
o[50(0,96)] r[71] $[45(0,93)] -t[108] E[70(0,102)] n[68]
-S[96] $[56] n[106(30,83)(100,83)] #
```

Figure 1: textual representation of an EPT for the sentence "Thank you for your attention"

An example of an EPT is provided by figure 1. The first value between square brackets is the phoneme duration (in ms), optionally followed by one or more intonation breakpoints. Each breakpoint consists of a location value (in ms) relative to the beginning of the phoneme, followed by a pitch value (in ST/4; reference 50 Hz).

A major asset of Prosody Transplantation is the combination of natural sounding speech with a low bit rate for storage (less than 300 bit per second). In addition, only the prosody and not the timbre of the speaker is retained. New donor messages can be recorded by new speakers and seamlessly integrated in existing applications. Specific tools have been developed to speed up the prosody transplantation process (Van Coile et al., 1994). Although the EPTs as such do not support linguistic variation, the combination of PTS with a template driven system provides linguistic flexibility as well as natural prosody.

3 The Message-to-Speech System

The message-to-speech system described in this section takes as input a message specification and outputs synthetic speech with highly natural prosody. Below, we first define the key concepts and then focus on two main modules of the system: the generation module and the prosodic module.

3.1 Key Concepts

A *message* can be seen as a complete sentence. It is specified as a concatenation of *message units* (building blocks that constitute prosodic units). The flexibility of a message unit is guaranteed by the presence of *slots*. A slot is a placeholder that can take an argument. A *carrier* is a template containing the enriched phonetic transcription of the canned text part, transplanted from an appropriate donor, and zero or more slots. For each slot, the carrier contains morpho-syntactic and prosodic information. By filling out a slot of a carrier with different arguments, several variants can be derived from the same message unit at run-time.

Figure 2 shows the wave and the prosody corresponding to the donor "in four miles". In order to obtain a flexible carrier, "four" is cut away and replaced by a slot in which any argument of the type /number/ (see figure 3) can be filled out at run time.

Figure 3 illustrates that the message "In four miles, bear left" is realised as a concatenation of two message units: "in */number/* mile*(s)*" and "bear left". The message unit "in */number/* mile*(s)*" has one slot in which a numeric argument is to be filled out. The message unit "bear left" has no slots.

message	In four miles, bear left
message specification	(message unit1 4) (message unit2)
message unit1	in */number/* mile*s*
message unit2	bear left
carrier 1	in */number/* miles #[952(952,101)]?[18]I[66]n[92(4,98)] /number: ...**ON=CO** .../ m[138(10,103)(70,96)] Y[224(2,93)(132,92)] l[173(58,82)] z[352] #[411(231,82)]
carrier 2	bear left # [50(1,124)] b[141(104,91)] E[228(211,119)] r[50]- l[60(4,120)] E[205(156,82)] f[131] t[151] #[800(800,79)]

Figure 3: example of message specification, message units and carriers for a message

3.2 Message-to-Speech Generation Module

The generation module translates each message unit of the message specification into a carrier with optional arguments. This translation is guided by a two-fold mechanism:

- argument dependent carrier selection consists in selecting a carrier in function of (a characteristic of) an argument. Figure 4 shows that the message unit "in */number/* mile*(s)*" can be realised by one out of two carriers, depending on the numeric argument that is filled out. If the argument is "1", the message unit is mapped on carrier 1a. In the other cases, the message unit is mapped on carrier 1b.
- carrier dependent argument realisation consists in determining the correct surface realisation of an argument, depending on properties of the slot in which it is inserted. Figure 5 illustrates that the argument "1" has a different surface

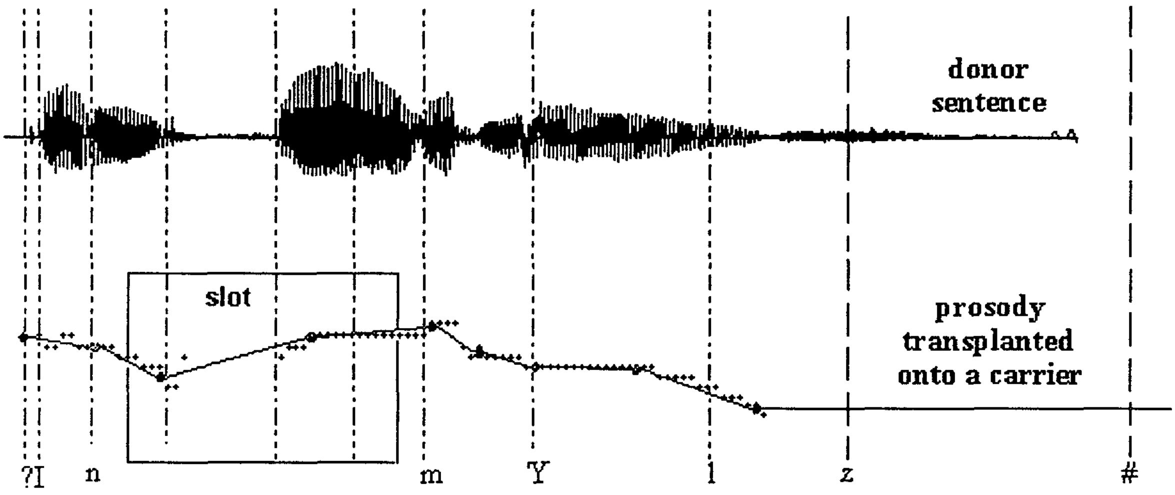

Figure 2: intonation contour for a carrier obtained from a donor sentence

message unit	mapping condition	carrier (represented orthographically)
"in */number/* mile*(s)*"	the argument = "1" the argument ≠ "1"	1a: "in /a/ mile_ 1b: "in /number/ miles

Figure 4: example of argument dependent carrier selection

realisation ("a" versus "an") depending on the phonetic on-set of the word to the right of the slot.

For the arguments filled out in the slot of a carrier, prosody is calculated at run-time (see section 3.3). As prosody derived from human recordings is preferred over prosody calculated at run-time, we try to keep the number of slots in a carrier as limited as possible. Therefore, the possibility is offered to delete arguments during the translation of message units into carriers. This functionality is exploited when the number of possible slot fillers is restricted. Figure 6 shows a message unit with one slot that is translated into one out of four carriers without slot, depending on the message unit argument.

3.3 Message-to-Speech Prosodic Integration Module

The purpose of the prosodic integration module is to calculate appropriate prosody for the arguments that are filled out in a slot of a carrier. Therefore, a phonetic transcription of the argument needs to be available. This transcription can be obtained by a dictionary look-up or by using a grapheme-to-phoneme conversion routine.

In a first step a duration is calculated for each of the phonemes in the argument. In a second step, an appropriate intonation contour is calculated.

3.3.1 Duration module

The input of the duration module is a phonetic transcription in which primary and secondary stress are indicated. The duration module has access to one or more duration models in order to produce a duration value for each phoneme in a phonetic transcription.

A duration model is a rule-based system calculating durations, taking into account parameters such as lexical stress, position of phonemes (word initial, word medial, word final, sentence final), length of the argument, phonetic context of phonemes (left/right neighbour, consonant cluster), etc. As speech rate can vary from one message to another, a slot specific speech rate coefficient, provided by the carrier, is also taken into account.

Two major strategies with respect to duration modelling can be discriminated:

- As the most natural prosody is the one derived from human speech, the possibility is offered to feed the duration module with phonetic transcriptions enriched with duration information copied from natural speech. When customising the MTS system, an argument dictionary containing this information can be built off-line

message unit	argument	surface realisation	condition
"in */number/* mile*(s)*"	1	a	word to the right of the slot has a consonantic on-set
"in */number/* hour*(s)*"	1	an	word to the right of the slot has a vocalic on-set

Figure 5: example of carrier dependent argument realisation

message unit with one slot	mapping condition	carrier (represented orthographically) without slot
"go to the */direction/*"	the argument = "west" the argument = "east" the argument = "north" the argument = "south"	go to the west go to the east go to the north go to the south

Figure 6: example of message unit argument deletion

by making use of the prosody transplantation tools (see section 2). If transplanted durations are available in the argument, they are taken over by the duration module and only modified in specific cases — e.g. change a duration in order to cope with a phenomenon such as final lengthening.

- For arguments without transplanted durations, a general purpose duration module is activated. It consists of a cascade of different duration models each having a decreasing specificity. Specific duration models exist for particular arguments such as numbers or date and time indications. The general purpose model is only used if a more specific model is not available. Special tools have been developed to speed up the creation of general and special purpose duration models.

3.3.2 Intonation module

The input of the intonation module is a phonetic transcription enriched with phoneme duration information. The output is a phonetic transcription describing both duration and intonation. After taking care of assimilation, this enriched phonetic transcription can be inserted without further action into the carrier.

There are two ways to model the intonation on arguments:

- The most natural intonation is obtained by transplanting part of an intonation contour as obverved in a donor sentence onto the argument that is to be filled out in a carrier. It is indeed possible to reuse the intonation as realised on "4.6" in the donor phrase "in 4.6 miles" for the argument "9.5" that is to be inserted in the carrier "in */number/* miles".
- If no appropriate donor contour is available, the intonation module calculates a piecewise linear intonation contour based on slot specific intonation models. Slot specific intonation parameters that are taken into account are among others the begin pitch, the end pitch, the declination rate and the intonation context (final fall, continuation rise, etc.).

4 Discussion

The Message-to-Speech system is designed to generate high quality speech output with the flexibility desired for spoken dialogue and message generating systems. It produces high quality speech while morpho-syntactic variations are taken into account. More specifically, as the message units and underlying carriers can take arguments, it is possible to generate several variants of the same basic message.

- variations on the level of a carrier slot can be paradigmatic: a message ranges over all the elements belonging to a certain semantic category (e.g. product name, cardinality, direction – see figure 6) but the actual message is not known on beforehand.
- variations on the level of a carrier slot can be syntagmatic: agreement of all kinds, liaison, contraction, etcetera (see figures 4 & 5).
- variations on the level of the message units can be semantic: new combinations of message units lead to the creation of new messages. E.g. the message unit "in */number/* mile*(s)*" can not only be combined with a message unit "drive */slowly_fast/*" but also with the message unit "bear */left_right/*".

Highly natural prosody for the carriers is obtained thanks to the prosody transplantation technique. The prosody transplantation technique can be used for the slot arguments as well. However, if no donor prosody is available for an argument, prosody is calculated at run-time on the basis of specific duration and intonation models.

5 Related Research

In what follows we try to relate the MTS system to the levels that are generally recognised to form part of a NLG system. A well known architectural scheme outlining the three basic levels of an NLG system has been proposed by Reiter (Reiter, 1994, p.164) [1]:

1. content determination and text planning: The content of the message to be communicated is mapped onto a semantic form, possibly annotated with rhetorical relations. On this level, reasoning takes place about the communicative goals of the text or message and the rhetorical relations between these goals.
2. sentence planning: The information of the semantic form is distributed over sentences and paragraphs. The sentences are linked together.
3. surface generation: The abstract specification of the linguistic structure is mapped to a surface form that communicates the information while syntactic and morphologic processing is done in order to generate a grammatically correct surface form.

If we compare our strategy with the classification proposed above, the mapping of a message unit onto carriers is to be situated on *the surface generation level*. The result of the mapping stage is a complete surface form (represented by an EPT): the precise wording of a message has been fixed in accordance with syntactic and morphologic restrictions. The prosodic integration phase has no explicit place in Reiter's architecture since he only studied text generation systems.

The content determination, text planning, and sentence planning levels are not provided by the MTS system. In a number of practical message generating systems, the content of a message corresponds in a straightforward manner with the message units, which can therefore easily be generated by the back-end application.

6 Current Developments

We are currently enhancing the functionality of the MTS system in the following areas:

- The MTS system in its current state only comprises carriers with one slot or multiple non-related slots. The slots of a carrier are filled in a fixed sequential way (left to right), so that the linguistic restrictions are also applied in the same order. This entails that no restrictions between related slots can be applied. Therefore, the selection mechanism risks entering a deadlock situation. E.g. consider the carrier "you have bought */number/ /item/*" where */number/* indicates the number of items */item/*. */number/* could be realised as "no, a(n), two, three" etc. In the case that "num = 1", the system blocks since the argument "1" cannot be realised as long as the phonetic on-set of the following word is not known. But that word cannot be realised (singular vs. plural) either since the number slot is not yet filled in.
- The MTS system in its current state only deals with atomic arguments. For some applications, it is also useful to support lists as arguments. A back-end application then could use the same message unit to have the MTS system generate e.g. "You have new mail from Tom" (atomic argument) or "You have new mail from Tom, Paul and John" (list argument). In order to achieve this, the MTS system will have to deal with syntactic aggregation (Dalianis and Hovy, 1996; Dalianis, 1996a).

References

Hercules Dalianis and Eduard Hovy. 1996. Aggregation in natural language generation. In Giovanni Adorni and Michael Zock, editors, *Trends in Natural Language Generation: An Artificial Intelligence Perspective*, pages 88–105. Springer-Verlag.

Hercules Dalianis. 1996a. Aggregation as a subtask of text and sentence planning. In J.H. Stewman, editor, *Proceedings of the Artificial Intelligence Research Symposium*.

Hercules Dalianis. 1996b. *Concise Natural Language Generation from Formal Specifications*. Ph.D. thesis, The Royal Institute of Technology and Stockholm University, Department of Computer and Systems Science, Stockholm, Sweden.

Michael Elhadad. 1992. *Using argumentation to control lexical choice: A functional unification-based approach*. Ph.D. thesis, Computer Science Department, Columbia University.

Eduard Hovy. 1995. Overview. In Ronald Cole, Joseph Mariani, Hans Uszkoreit, Annie Zaenen,

[1] A more recent and detailed description can be found in (Reiter and Dale, 1997).

and Victor Zue, editors, *Survey of the State of the Art in Human Language Technology*, pages 161 - 169. Cambridge University Press (in press).

Ehud Reiter and Robert Dale. 1997. Building applied natural language generation systems. *Journal of Natural Language Engineering*, pages 1–38 (submitted).

Ehud Reiter, Chris Mellish, and John Levine. 1995. Automatic generation of techical documentation. *Applied Artificial Intelligence*, 9(3):259–287.

Ehud Reiter. 1994. Has a consensus nl generation architecture appeared, and is it psycholinguistically plausible? In *Proceedings of the Seventh International Workshop on Natural Language Generation*, pages 163–170, Nonantum Inn, Kennebunkport, Maine, June 21-24,.

Ehud Reiter. 1995. NLG vs. templates. In *Proceedings of the European NLG Workshop 95*, pages 95 - 106.

Harold Somers, Bill Black, Joakim Nivre, Torbj on Lager, Annarosa Multari, Luca Gilardoni, , Jeremy Ellman, and Alex Rogers. 1997. Multilingual generation and summarization of job adverts: the TREE project. In *Proceedings of the Fifth Conference on Applied Natural Language Processing*, pages 269 - 276, Washington D.C. Morgan Kaufmann Publishers.

B. Van Coile, L. Van Tichelen, A. Vorstermans, J.W. Jang, and M. Staessen. 1994. Protran: A prosody transplantation tool for Text-to-Speech applications. In *Proceedings of the 1994 International Conference on Spoken Language Processing (ICSLP-94)*, pages 423–426, Yokohama, Japan.

B. Van Coile, H. Rühl, L. Vogten, M. Thoone, S. Goss, D. Delaey, E. Moons, J. Terken, J. de Pijper, M. Kugler, P. Kaufholz, R. Krüger, S. Leys, and S Willems. 1995. Speech synthesis for the new pan-european traffic message control system RDS-TMC. In *Proceedings of Eurospeech 1995*, pages 145–148.

K. van Deemter and J. Odijk. 1997. Context modeling and the generation of spoken discourse. *Speech Communication*, 21:101 - 121.

K. van Deemter, J. Landsbergen, R. Leermakers, and J. Odijk. 1994. Generation of spoken monologues by means of templates. In L. Boves and A. Nijholt, editors, *Proceedings of the Eight Twente Workshop on Language Technology*, pages 87 - 96, Twente.