A compact Representation of prosodically relevant Knowledge in a Speech Dialogue System

Peter Poller
German Research Center for
Artifical Intelligence (DFKI GmbH)
Stuhlsatzenhausweg 3
66123 Saarbrücken, Germany
poller@dfki.uni-sb.de

Paul Heisterkamp
Daimler–Benz AG
Forschungszentrum Ulm
Wilhelm–Runge Str. 11
89081 Ulm, Germany
heisterkamp@dbag.ulm.daimlerbenz.com

Abstract

The acceptance of speech dialogue systems by the user is critically dependent on the degree of "naturalness" realized. The speech generation and synthesis modules have to be able to run in real time and to produce high–quality speech output. To produce naturally sounding speech, the synthesizer has to have not only the knowledge of the words to utter and the order in which they appear but also information about their structural relationship. The latter is expressed acoustically in the form of prosody, i.e. how the voice raises and falls during an utterance, the rhythm, where pauses are set, etc. Prosody is also influenced by the properties associated with given words in the context of an utterance, e.g. the focus of a sentence or certain emphatic elements. This article describes a compact representation for conveying this type of information from the generator to the synthesizer in a modular system and describes how (parts of) this information is (are) derived in the EFFENDI system, the generation module for a speech dialogue system for train inquiries.

1 Introduction

The speech generation and synthesis modules of a speech dialogue system are very important, since they form the output "visible" to the user. Natural-language generation for a speech dialogue system must therefore operate in real time. In a system which outputs speech, real time is essentially the time the system takes before it starts to utter what it has to say. The reaction time to the previous user input should be minimal. One way of increasing throughput, and thus coming closer to real-time operation, is to divide the input into small autonomous packets which can be processed independent of each other and thus simultaneously. Then, once the first part of an utterance has been generated, it can be passed on to the synthesis module, and while the latter is producing speech for this segment, the generator can proceed to process the next segment. This type of processing is known as *incremental processing*.

The acoustic speech signal is by its nature volatile; it can only be heard once. This makes it imperative that the generator provides the synthesizer with all information necessary to produce high-quality speech. In human speech, information is conveyed not only in the individual words, but also in the prosody which provides the listener in many subtle ways how these words are related to each other. It often conveys information not explicitly contained in the spoken words, such as the focal point of a sentence or the contrast of certain words (ideas) with other spoken or unspoken words. A prerequisite for high-grade synthetic speech is therefore that all phonologically and prosodically relevant information contained in the concepts forwarded to the generator and the syntactic structure produced by it be passed on in a suitable form to the synthesis module. Only the consideration of structural information can assure the production of high-quality prosody by the synthesis module. The details of these "phonological" structures are of course language dependent, but there are certain properties common to all languages. As the overall system described here is strictly modular and (in principle) multilingual, the phonologically relevant information of an utterance has to be coded in a high-level *interface protocol*, that can both be output by a variety of generators as well as used as input for a variety of synthesis modules.

Since the design of such an interface protocol depends on the structure of the semantic input concepts and the syntactic structures generated from them, section 2 gives a short overview of the dialogue management module and our generation system that has been developed in the EFFENDI project[1]. Section 3 then describes the compact representation for prosodically relevant knowledge and briefly indicates how the information for this representation is obtained from the input concepts of the generator and the syntactic structures generated from them. As incremental generation often requires the repair of some previously generated parts, section 4 considers the effects of incremental generation that concern the synthesis module and how we try to avoid unnecessary repetitions of words as far as possible. The final section describes the goals of the on-going work in EFFENDI.

2 Overview of the Dialogue System

The syntactic generator EFFENDI is integrated into the speech dialogue system implemented by Daimler-Benz. The generator itself is particularily adapted to the specific needs of a real time speech dialogue system (cf. (Poller and Heisterkamp 1997)). A more detailed description of the diaogue system as a whole can be found elsewhere (cf. e.g. (Brietzmann et al. 1994), (Hanrieder and Heisterkamp 1994) or also (Heisterkamp 1993)). We will thus restrict our description to those components of the dialogue system that interact with the generator.[2]

The planning of a system utterance (also called "strategic generation" or "what-to-say") is the main task of the dialogue management component. This means the determination of the appropriate type of utterance in a given dialogue situation, the items that are to be talked about in which manner or style and finally to deliver a semantic description of the utterance to the syntactic generator.

A module called the *Dialogue Manager* operates with a set of goals (cf. e.g. (Heisterkamp and McGlashan 1996)) that result from the contextual interpretation of the user utterance in a *Belief Module* ((Heisterkamp et al. 1992)), the requirements of the application system, and the current confirmation strategy (cf. (Heisterkamp 1993)).

The *Dialogue Module* selects from the overall set of goals that subset which should constitute the next system utterance. A *Message Planner* receives this subset consisting of types utterances (e.g. a request for confirmation), the task item of that goal (e.g. a departure place) and the status of this item (new, repeated n times). It requests a semantic description of that task item from the *Belief Module.* The semantic description is then combined with the dialogue goal types for the phrase type markers (e.g. question) and verbosity markers inferred from the status (e.g. the possibility of ellipting a verb or reducing it to a prepositional phrase) to result in a semantic structure[3]. This semantic structure is then passed on to the generation module. A special interface translates these semantic representations into syntactically oriented input specifications for the generator.

The most important property of the EFFENDI generator is its incrementality. Incremental generation means that both the consumption of the input elements as well as the production of the output elements work in a piecemeal and interleaved fashion. Input consumption and output production interleave in such a way that first parts of a sentence are uttered before the generation process is finished and even before all input elements are consumed. This kind of flexible syntactic generation is only possible if the processing can be broken down into a large set of independent tasks which can run in parallel (cf. (Kempen and Hoenkamp 1982)). Applying this principle, generation in EFFENDI is realized by synchronizing a set of actively communicating, independent processes (so-called *objects*) each of which is responsible for the syntactic realization of an input element and its integration into the syntactic structure of the whole utterance (cf. (Kilger 1994)).

In addition, incremental generation should be separated into two main computational steps. The first step must comprise the construction of the hierarchical (syntactic) structure. The word order of the surface string is computed in a second step (linearization). The reason for this separation is the observation that decisions at the hierarchical level are often possible at a time where input information is not yet sufficient to make decisions at the positional level ((Kilger 1994)).

Incremental syntactic generation can therefore be organized as follows. The incremental input interface immediately translates each incoming input specification into an independent process (*object*).

[1]EFFENDI stands for "EFfizientes FormulierEN von DIalogbeiträgen" (Efficient formulation of dialogue contributions) and is a joint research project of the DFKI Saarbrücken and Daimler-Benz Research Ulm.

[2]Historically, our dialogue system goes back in part to the one developed in the SUNDIAL project. The architecture of that system was laid out to accommodate a generator (cf. (Youd and McGlashan 1992), but for various reasons the work on this aspect was discontinued.

[3]The planning process also has access to knowledge about recency, semantic focus etc. This knowledge is incorporated in the planning result.

This process immediately and independently runs the following computational steps. At the hierarchical level, an elementary syntactic structure for the individual input element is selected. In order to build a virtual syntactic structure for the whole sentence, the objects exchange structural and syntactic information by explicitely sending messages to related objects. An object that completes the structural combination with related objects, changes to the positional level the task of which is the determination of the resulting word order of the surface string (*linearization*) and its output. Linearization and output production have to be synchronized with respect to the word order that globally results from the local linearizations. So, incremental output production is organized as a global visit of all objects. As soon as an object has finished its linearization, it can be uttered, i.e. sent to the synthesizer. The incrementality of the output is automatically ensured because the individual objects finish their local linearizations at different times.

3 The Interface Protocol

The goal of the interface protocol is to form a compact representation that contains all phonologically and prosodically relevant information which the generator can currently derive from its *concept* input or the syntactic structures generated from it. Both are relevant for phonological realisation, but they neither are nor directly contain the phonological knowledge itself, as the strategic generation, linguistic generation and final synthesis task are divided into different modules. The representation concerns categories that the prosodic construction makes use of rather than instructing it directly.[4]

A *phonologically* oriented description suitable for generating proper sentence prosody differs in many aspects from the traditional *syntactically* oriented description normally produced by a sentence generator such as EFFENDI. The following section shows how the basic phonological specification can be derived from existing semantic and syntactic structures of an utterance in three main steps. For reasons of simplicity the treatment of incremental processing is postponed to the next section.

[4]In integrated systems, where conceptual construction, generation and synthesis have full mutual access to the relevant knowledge, there is no need for such an interface, and the linguistic grammar can directly incorporate the phonological features (cf. e.g. (Prevost and Steedman 1994)). However, apart for lack of flexiblility, integrated systems mostly *must* make use of the concept-to-speech synthesis ((Steedman 1996)), whereas the interface presented here can also be used with a text-to-speech synthesis.

3.1 Phonological Categorization

In classical grammars every word belongs to a category which describes how words of this category may be inflected and how they interact with other words in a sentence on both a syntactic and a semantic level. In formal computer grammars for parsers and generators, words are also assigned to categories. We will call these categories and all other phenomena connected with such grammars "syntactical."

The structures necessary to describe the prosodic behavior of a sentence or utterance may differ considerably from those necessary for classical grammars. In this paper we refer to all phenomena associated with prosodic or pronunciational behavior, as opposed to that described above, as being "phonological". In this sense each word to be uttered has a *phonological* category associated with it. These categories tell the synthesizer something about the phonological function of each word in a sentence, in particular about the relative stress of the words to be uttered. These categories will often differ from the purely *syntactic* categories, which define the semantic and syntactic function of each word in a sentence. These categories will vary from language to language. In addition to the phonological category, one or more special attributes such as focus or emphasis (coming from the semantic generator input) may be optionally associated with each word.

3.2 Phonological Segmentation

In every language, spoken sentences are broken up into so-called "thought groups" or "breath groups" if they are more than a few words long. Also certain "atomic" groups such as "in the big room" are never broken up any further. The elements that constitute an atomic group are of course language dependent. These phonologically oriented atomic groups may or may not correspond to syntactic groups (i.e. subtrees) produced by the generator, but can be derived from the latter. Each atomic group also has a group category associated with it, which describes how the group interacts prosodically with others. Some of the group categories we initially propose for German are summarized in the following. Note that, e.g., "phonological" conjunctional phrases have no phrasal counterpart on the syntactic level:

SP (Subject Phrase)

> **Example:** **Der Zug** fährt nach Ulm.
> **Definition:** A noun phrase or a pronoun used in the nominative case

PP (Prepositionl Phrase)

> **Example:** Der Zug fährt **nach Ulm.**
> **Definition:** A prepositional phrase

AP (Adverbial Phrase)

> **Examples:** **morgen früh**; Der Zug fährt **jeden Tag**.
> **Definition:** One or more adverbs or an adverbially used noun phrase

KP (Conjunctional Phrase)

> **Examples:** über Ulm **und München**; ..., **weil der Zug** nicht fährt.
> **Definition:** A conjunction together with the following syntactic segment

V (Verb)

> **Example:** Der Zug **fährt**.
> **Definition:** An isolated inflected verb in a main clause

VP (Verb Phrase)

> **Examples:** ..., ob man **fahren kann**; ..., ob der Zug **fährt**
> **Definition:** A complete verb phrase if all words are contiguous or an isolated inflected verb in a subordinate clause

3.3 Association of Atomic Groups to Each Other

Once the atomic groups have been determined, it is necessary to specify how these groups are logically connected to each other. In a phrase such as

$\boxed{\text{from the man}}\ \boxed{\text{in the room}}$
$\boxed{\text{wearing the coat}}$

the second atomic group is logically connected to the first because "*in the room*" refers to *man*. Likewise the third group is also logically connected to the first group rather than its antecedent because "*wearing the coat*" also refers to *man* and not to *room*. This type of information can be derived from the original syntactic tree structure produced by the generator module. How such groups are connected to each other has a bearing on how the ultimate division into breath groups is determined by the synthesizer module.

3.4 The Protocol

This section describes the formal syntax of the interface protocol and illustrates it with an example. Each interface protocol describes a dialogue *turn*, which may consist of one or more sentences. In our example the turn consists of a single sentence. The protocol contains the following information:

- the type of each sentence to be uttered,
- a list of all words to be uttered along with their associated categories, special attributes if any, and the order in which they are to be uttered,
- a specification of each atomic group along with its associated group category and
- a description of all logical connections between atomic groups.

The interface protocol for the sentence "Sie möchten wissen, wann der Zug nach Ulm fährt" (literally: You would like to know, when the train to Ulm leaves.) looks like this:

```
$AS
** Sie(PRON)                           #SP   >+1
** möchten(H) wissen(VU)               #VP   >-1 >+1
** wann(KONJ) der(DET-S) Zug(N)        #KP   >-1 >+2
** nach(PRAEP) Ulm(N)                  #PP   >+1
** fährt(V)                            #VP   >-2 >-1
```

Each sentence of an interface protocol consists of a specification of the sentence type, followed by a description of the atomic groups in the order they are to be uttered. The sentence-type descriptor is uniquely identified by the initial "$" and also serves to separate sentences from each other. Currently, the follwing types of sentences are distinguished:

- **$AS** — Affirmative proposition
- **$WF** — Wh-question
- **$JNF** — Yes/No-question
- **$BS** — Imperative clause

Each atomic group is introduced by "**", after which the individual words of the group along with their category and any optional attributes are listed. Word categories are enclosed in parentheses. Attributes, if present, are enclosed in square brackets, such as "`[focus]`" to indicate that the word in question forms the sentence focus. The last word/category pair is then followed by the group category, which is uniquely identified by the preceding "#". Finally a series of one or more pointers specifies other groups that are logically related. Each pointer is introduced by a ">" followed by a signed number which specifies how many groups before (-) or after (+) the present group the connected group lies. These pointers are effectively double headed.

In the example the first group points to the second (>+1), and the second group points back to the first one (>-1). This protocol is designed in such a way that all spacing between elements, as shown in the example, is optional.

Apart from the use in EFFENDI, the protocol is also used as synthesis input specification in the *VERBMOBIL*[5]-project ((Wahlster 1993)), for the system utterances within the german clarification dialogue.

4 The interface to the synthesis component in EFFENDI

This section considers the question how the interface protocol can be used when the syntactic generator and the synthesis module interleave incrementally meaning that some words of the output are handed over to the synthesis module while others are still being generated at the same time. The problem for this processing mode is that the pieces handed over to the synthesis module cannot contain all prosodically relevant information as far as sentence parts that have not yet been generated are concerned. In sequential processing the complete protocol for an utterance is automatically computed and handed over to the synthesis module in one single step. In incremental processing the protocol must be handed over to the synthesis module in a piecemeal fashion. The question is therefore how the information handed over to the synthesis module can be reduced in favor of an early beginning of the articulation of a system answer.

Since the protocol consists of a separation into breath groups, it seems to be reasonable to hand them over to the synthesis module as soon as they have been identified[6]. In order to minimize the number of missing pointers it is possible to impose a delay on one or more breath groups. This means that a breath group is handed over to the synthesis component if some of the following breath groups have already been identified by the generator.

The most important problem in incremental generation is the necessity of repairs that have to be done if, e.g., a previously unknown word cannot be attached to the word order already articulated. Since already articulated words cannot be retracted, an extensive repetition of the concerned phrase is necessary to correct the already articulated but wrong formulation. E.g., if the noun phrase "the man" has been articulated and it is incrementally extended by an adjective "young", the correction of the articulation consists of the repetition of the whole phrase "the young man".

In order to avoid such extensive repetitions, we developed a strategy called "afterthought syntax". If words resulting from semantic information that was not available when the first words of a sentence were uttered can't be syntactically correctly attached to the words already articulated, then the syntactic ordering is (partly) disregarded, i.e. precedence is given to completeness of the semantic content and shortness of the utterance over syntactic correctness. In virtually all cases, the resulting utterance remains completely understandable. Technically this behaviour is implemented using elliptic generation. The (now complete) utterance is regenerated, and all parts of the utterance that have already been uttered are marked as ellipses, i.e. prohibited from being uttered again. However, rules are applied to ensure that repair elements receive a syntatic context if they need it, thus overriding that prohibition, if necessary:

`Sie möchten wissen, wann der Zug fährt ...`
(You want-to know, when the train leaves ...)

`der nächste Zug ...` (the next train ...)

`nach Ulm.` (to Ulm.)

The first elliptical resumption is caused by the previously unknown adjective "nächste" which leads to the repetition of the complete noun phrase, while the second resumption is caused by the PP "nach Ulm" which, according to standard German syntax, would have to be placed before the verb in a subordinate clause.

5 Future Work

For the near future, we plan to implement a full interaction between the dialogue manager, the genera-

[5]VERBMOBIL is a translation system that can assist a face-to-face dialogue between two non-native english speakers. The dialogue partners have the option to switch to their repective mother tongue and to activate VERBMOBIL to translate their utterances into english. This processing sometimes requires a clarification dialogue, e.g., if some background noises irritated the recognizer.

[6]Note, that the identification of the breath groups runs in parallel to the ongoing generation, so that the only missing information may be some pointers to breath groups that have not yet been generated.

tor, and the speech synthesis module in incremental processing. We hope to gain practical experience in interleaved generation and synthesis. This is especially vital for finding an answer to the question, how articulation can be delayed in favor of an acceptable output quality in such a way that the overall reaction time of the system is only marginally increased.

References

A. Brietzmann, F. Class, U. Ehrlich, P. Heisterkamp, A. Kaltenmeier, K, Mecklenburg, P. Regel-Brietzmann, G. Hanrieder, W. Hiltl. 1994. Robust speech understanding. In *Proceedings of ICSLP-1994*, Yokohama, 1994.

G. Hanrieder, P. Heisterkamp. 1994. Robust analysis and interpretation in speech dialogue. In: *H. Niemann, R. de Mori, G. Hanrieder (eds.): Progress and prospects of speech research and technology, Proceedings of the CRIM/FORWISS workshop*, Munich 1994.

P. Heisterkamp. 1993. Ambiguity and uncertainty in spoken dialogue. In: *Proceedings of EUROSPEECH '93*, Berlin, 1993.

P. Heisterkamp, S. McGlashan, N. Youd. 1992. Dialogue semantics for an oral dialogue system. In: *Proceedings of ICSLP-1992*, Banff, Alberta, Canada, 1992.

P. Heisterkamp, S. McGlashan 1996. Units of dialogue management. An example. In: *Proceedings of ICSLP-1996*, Philadelphia, PA, 1996.

G. Kempen, E. Hoenkamp. 1982. Incremental Sentence Generation: Implications for the Structure of a Syntactic Processor. In: *J. Horecky (ed.), 9th International Conference on Computational Linguistics*, 1982.

A. Kilger. 1994. Using UTAGs for Incremental and Parallel Generation. In: *Computational Intelligence, 10 (1994) 4*, pp. 591–603, 1994.

J. Peckham. 1993. A new generation of spoken dialogue systems: Results and lessons from the Sundial project. In: *Proceedings of EUROSPEECH '93*, Berlin, 1993.

P. Poller, P. Heisterkamp. 1997. Hybrid Knowledge Sources for Generation in a Speech Dialogue System. submitted to ACL/EACL 1997, Madrid, Spain.

S. Prevost, M. Steedman. 1994 Specifying Intonation from Context for Speech Synthesis. In: Speech Communication, 15, pp. 139–153, 1994.

M. Steedman. 1996 Representing Discourse Information for Spoken Dialogue Generation. In: *Proceedings of International Symposium on Spoken Dialogue (ISSD '96)*, pp. 89–92, Philadelphia, PA, 1996.

W. Wahlster. 1993. VERBMOBIL: Translations of Face–to–Face Dialogs. In *Proceedings of the MT Summit IV*, Kobe, Japan, pp. 127–135, 1993.

N. Youd, S. McGlashan. 1992. Generating Utterances in Dialogue Systems. In: *R. Dale, E. Hovy, D. Rösner, O. Stock (Eds.) (1992): Aspects of automated natural language generation. Proc. of the 6th International workshop on natural language generation*, Trento, Italy, April 1992.