Context Modeling and the Generation of Spoken Discourse

Kees van Deemter and Jan Odijk

Institute for Perception Research (IPO)
P.O.Box 513, 5600 MB Eindhoven, The Netherlands.

E-mail: {deemter,odijkje}@natlab.research.philips.com

Abstract

This paper presents the Dial-Your-Disc (DYD) system, an interactive system that supports browsing through a large database of musical information and generates a spoken monologue once a musical composition has been selected. The paper focuses on the generation of spoken monologues and, more specifically, on the various ways in which the generation of an utterance at a given point in the monologue requires modeling of the linguistic context of the utterance.

Zusammenfassung

In diesem Artikel wird das Dial-Your-Disc (DYD) System vorgestellt. Dieses interaktive System unterstűtzt den Benutzer beim Durchsuchen einer grossen Datenbank mit Informationen űber Musikkompositionen. Nach Auswahl einer Komposition erzeugt das System einen gesprochenen Monolog. Dieser Artikel konzentriert sich auf die Erzeugung gesprochener Monologe. Insbesondere wird besprochen, in welcher Weise die Erzeugung einer A̋usserung je nach ihrer Position im Monolog es erfordert, den linguistischen Kontext der A̋usserung zu modellieren.

Résumé

Cet article présente le système DYD (Dial-Your-Disc/Choisissez votre disque), qui comprend entre autres la fonction de recherche dans une grande base de données, et celle de génération d'un monologue sous forme vocale, une fois que l'oeuvre musicale a été sélectionnée. L'article présente en détail la génération des monologues, avec un accent particulier sur la nécessité de recourir au contexte linguistique de l'énoncé pour effectuer différemment la génération en fonction de la position courante dans le monologue.

Keywords: language/speech generation; music information system; accent location; prosody; context modeling; discourse structure.

1 Introduction

Context modeling is becoming an increasingly important ingredient of linguistic theories and applications.

Probably, the area of linguistics in which the modeling of contextual information has been studied in most depth is that of language *interpretation.* For example, it has long been recognized that deictic words such as *this* and *now* depend for their interpretation on the context in which they are uttered (see, e.g., (Montague 1974, Kaplan 1979) for formal treatments). Likewise, the interpretation of personal pronouns has long been known to be dependent on the linguistic environment in which they appear. Moreover, it has become clear that similar kinds of dependence on linguistic context apply to many other classes of expressions.[1] Inspired by this growing body of empirical work, dependence on linguistic context has become the cornerstone of the so-called dynamic theories of meaning (Kamp and Reyle 1993, Heim 1982, Barwise 1985, Groenendijk and Stokhof 1991). These theories characterize the meaning of a sentence as its potential to change one 'information state' into another, and it is this dynamic perspective on which current natural-language interpreting systems are beginning to be based. Thus, contextual interpretation, far from being viewed as an exception, is taking centerstage in the theatre of language interpretation.

The importance of context for language interpretation raises the question in how far context modeling is relevant for natural-language generation (NLG). Of course, the relevance of contextual information for generation follows from the relevance of this information for interpretation, since the discourse generated has to be interpreted by the user of the system. For example, when a pronoun or a deictic expression is generated, its linguistic context determines how it will be interpreted by the user. Consequently, it is hard to see how an NLG system can generate texts that make a correct use of pronouns, deictic expressions, etc., without modeling the kinds of context that their interpretation depends on.

But there is more. What has been said at a given point in the discourse, and the way in which it has been said, help to determine both what can be said afterwards and what is the best way to say it, as will be pointed out later in this paper. For example, it seems reasonable to require that each sentence in a discourse contributes novel information, or in other words, information that is not already present in the preceding sentences. Furthermore - and this is where the relevance of context for *speech* comes in - those parts of a sentence that are responsible for novel information are likely to be accented in speech. Thus, both content and form of *any* utterance are affected by linguistic context.

We will try to clarify some of the various ways in which linguistic context is relevant for the generation of spoken monologues and, more specifically, how the various kinds of contextual information are modeled in a system that generates spoken texts. First, we will outline the system that has been the vehicle for our work in discourse generation, namely the so-called Dial-Your-Disc (DYD) system (section 2). The next two sections outline the

[1] Examples include the contextual interpretation of full Noun Phrases (Sidner 1979, Carter 1987, van Deemter 1992); implicit arguments (Partee 1989, Condoravdi and Gawron 1996); tense (Hinrichs 1986); vagueness (Klein 1980); and presupposition projection (Heim 1992, van der Sandt 1992).

language-generation module (section 3) and the prosodic module (section 4), respectively. The penultimate section draws some conclusions (section 5) and the Appendix exemplifies the output of the DYD system (section 6). These monologues are presented in written form, but with annotations that mark the locations of accents.

2 The DYD system

Speech generation is now generally regarded as a technology with commercial possibilities (see, e.g., (Cole *et al.* 1995)). Possible applications include various 'hands busy, eyes busy' applications such as assistance in car navigation, as well as, for example, telephone-based information services. In the latter type of applications, speech is typically used for the presentation of information stored in the formal representations of a conventional database and that a user has somehow expressed interest in. The DYD system, developed at the Institute for Perception Research (IPO), is an example of a system of this kind. It produces spoken monologues derived from information stored in a database about compact discs with compositions written by W.A.Mozart. The database contains information about the compositions, the performers, and the recordings. In DYD, the need for spoken monologues arises after a user has expressed his or her interest in a particular kind of musical composition. The monologues produced are interleaved with musical fragments. A system of this kind could be used for teleshopping, but it could also be used for educational purposes or just for fun. In all of these cases operating the system should be easy and pleasurable. In particular, this implies that the monologues should be lively.

In the present version of the system, users can express their interests through an extremely simple, mutilated form of typewritten or spoken[2] English, in which they can accumulate properties that they like to hear in a composition. For example,

> User says *sonata* - The system selects the set of tracks that are part of a sonata
> User says *piano* - The system selects tracks that are part of a sonata and contain a piano.
> User says *fast* - The system selects tracks that are part of a sonata, contain a piano, and have tempo indication 'allegro' or 'presto'. (See also Appendix.)

To discard old properties, the user can ask for a *reset*. When a property is added that is incompatible with the previous requests, the system will either drop one or more older properties, or explain that it has arrived at the empty selection. In any case, the system will produce a simple 'feedback' sentence to characterize the selection it has come up with. For example, *I have selected 54 tracks with tempo allegro or presto.*

Note that there is a clear distinction between 'browsing'-oriented systems of this kind, on the one hand, and traditional question-answering systems, which give precise answers to precise questions, on the other. The exact way in which users can indicate their areas of interest will not be discussed in this paper, and the same is true for the ('selection characterizing') feedback that the system produces. Instead, the present paper focuses on the situation that

[2] In speech-recognition mode, DYD uses word-spotting to recognize key words, such as *sonata*, *piano*, *fast*, etc. The speech-recognition module makes use of the PHICOS recognizer generator developed at Philips Research Laboratories in Aachen (see, e.g., (Ney *et al.* 1994)).

arises after this feedback is given, when the system makes a random choice from among the tracks in the selection and provides more elaborate information on the track selected. Suppose the system focuses on the fourth part of a particular recording of Mozart's composition K309. A database representation of this recording could be:

KV 309
DATE 10/1777 - 11/1777
SORT piano sonata
NUMBER 7
PERFORMER Mitsuko Uchida
PLACE London
VOLUME 17
CD 2
TRACK 4

DYD can then generate many alternative descriptions of this database object, one of which is chosen. Examples of such descriptions can be found in the Appendix.

Since liveliness is such an important requirement of the DYD system, it is essential that a large variety of texts can be produced expressing a given piece of information. Presentations are generated on the basis of database information by making use of syntactic *templates*: structured sentences containing open slots for which expressions can be substituted. These templates indicate how the information provided by a database object can be expressed in natural language. The required variety is achieved by having many different templates for the same information and by having a flexible mechanism for combining the generated sentences into texts. Information that does not fit in the uniform database format is called *gossip*, and is represented by object-specific templates expressing this information.

One condition that a template must fulfill if it is to be used at a certain point of the monologue is that there is enough information in the database to fill its slots. However, there are other requirements and these have to do with the linguistic context of the template. Some of these concern one specific way in which the gaps in the template are filled, while others have to do with the template itself. An example of the first kind occurs when the appropriateness of a pronoun depends on whether the preceding sentence contains a suitable antecedent for it. An example of the latter kind occurs when it is required that a template deals with the same topic as the other sentences in the paragraph.

We have seen that the linguistic context of an utterance consists of all the utterances that precede it, up to and including the utterance itself. The preceding utterances can occur in the same monologue, or at an earlier phase of the interaction between system and user. *Modeling* the context of an utterance requires finding a level of representation that is both rich and explicit enough to allow a system of *rules* to exploit the information in it in such a way that contextually appropriate utterances can be generated. Thus, context modeling comprises (1) setting up a data structure and filling it with information, and (2) formulating rules that exploit this data structure. The data structure will be called the Context Model.

It might be thought that Context Models of the kind we require are already available, given that the formal modeling of linguistic discourse has taken up so much of the work in linguistics of the last 15 years. It is true that the so-called Discourse Representation Structures (DRSs) of Discourse Representation Theory (DRT), for example, represent the contextual information that is relevant for the interpretation or generation of anaphoric (e.g., pronominal) material in a discourse (Kamp and Reyle 1993). But, these structures contain both less and more than what is required for generation. For example, as will become clear later, context models for generation have to contain, for a given sentence, a representation of its subject matter (its 'topic', see section 3.2), of the templates used in it, and of the degree of explicitness with which the sentence provides certain bits of information. Conversely, conventional DRSs (e.g., (Kamp and Reyle 1993)) contain plenty of semantic information that is not immediately relevant for current (i.e., generative) purposes. For these reasons, DYD has not implemented DRT or any other existing theory of linguistic context. Instead, we have created a data structure containing all and only the information that is needed for the purposes of generation. Four data structures are distinguished containing four different kinds of information extracted from all preceding dialogues and monologues, which are respectively called Knowledge state, Topic state, Context state, and Dialogue state. These structures will be motivated and discussed in the following section.

Figure 1 shows an overview of the architecture of the DYD system. The module called *Generation* generates syntax trees on the basis of the database and a collection of templates, and it maintains the context model. *Prosody* uses the Context Model to transform a syntax tree into a sequence of annotated words, where the annotations specify accents and pauses. *Speech* transforms a sequence of annotated words into a speech signal.

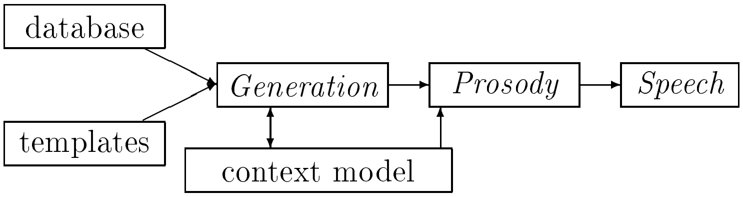

Figure 1: System Architecture: *Generation* generates syntax trees on the basis of the database and a collection of templates, and it maintains the Context Model. *Prosody* operates on syntax trees and the Context Model to produce a sequence of annotated words.

Most of the DYD software is written in the programming languages C and C++, but the language generator is written in a Philips-internal high level programming language called *Elegant*, which compiles into C (Augusteijn 1990). The application runs on a SUN SPARC-10 and makes use of a software-controllable CD jukebox to select and play music fragments. Text generation in the DYD system is usually instantaneous, and speech output is continuous. We believe that scaling up the system will be possible without any real efficiency problems.

3 Context in discourse generation

In this section we will describe how coherent monologues are generated in DYD.

Every system that generates texts has to contain rules to constrain the ways in which well-formed sentences can be combined into well-formed texts. Such systems may, for example, use a discourse grammar to specify, in the fashion of a sentence grammar, which successions of sentences are well-formed. (One example of a grammar of this kind is Rhetorical Structure Theory, (Mann 1987).) Alternatively, or in combination with a grammar, NLG systems often use a planning module, to guide the choice of referring expressions, the aggregation of information, and so on. DYD, by contrast, does not make use of a discourse grammar, nor does it contain a separate planning module. Instead, it uses (if we disregard, for this moment, a number of conditions that prevent the generation from sentences that deal with the wrong subject matter, cf. sections 3.2. and 3.3) a generate-and-test strategy in which the testing part contains elements of both discourse grammar and planning. For example, if, at a given point of the discourse, a sentence is generated that contains an uninterpretable pronoun, then this would-be continuation of the discourse is refuted, and another sentence is attempted. This global set-up was chosen because it guarantees a highly varied output: Even initial versions of the system contained a great variety of sentence and discourse structures. Of course, some of the structures generated are flawed, but the errors point the way to new conditions (i.e., stricter tests), and this has proved to be an important heuristic advantage of the 'generate-and-test' method.

Section 3.1 introduces *syntactic templates* and describes how the basic ingredients of monologues, i.e. individual sentences, can be generated from templates in the DYD system. Section 3.2 describes how templates 'string together', so to speak, to form coherent monologues. Section 3.3 describes the role of the Context Model in determining whether the conditions that a template must satisfy in order to be applicable hold. Section 3.4 specifies the role of the Context Model with regard to the discourse-syntactic and -semantic conditions which candidate sentences generated from a template must satisfy. Before we go on, however, we will first give a brief overview of the four submodels adopted in the actual implementation. The four submodels are: the Knowledge state, the Topic state, the Context state, and the Dialogue state. The *Knowledge* state keeps track of what information has been conveyed, and when. The *Topic* state keeps track of which topics have already been dealt with and which are still to be dealt with. The *Context state* (one part of the Context Model) keeps track of (i) the objects introduced in the text and their location, and (ii) the linguistic means used in the preceding text. Finally, the *Dialogue* state keeps track of which recordings were selected before, what the current recording is, and what kind of monologue should be generated. All of this will be discussed in more detail below.

For ease of reference we have put these submodels in table 1. The table makes an effort to cluster elements of the Context Model that belong together, but a certain arbitrariness is unavoidable. Not all elements occurring in the table will be clear at this point, but their role will be made clear in the rest of the text.

Submodel	Contains
Knowledge state	what, when, how told
Topic state	todotopics, donetopics, attemptedtopics, current topic
Context state	D-model, templates/words used, recently used concepts, location, alwaysgiven
Dialogue state	recording (part) selected; history of previous selections, whattotell

Table 1: Parts of the Context Model and the information they contain.

3.1 Syntactic Templates

The basic ingredients for discourses are sentences. So there must be a mechanism to generate sentences. Sentences are generated from *syntactic templates*, i.e., syntactic trees containing variable parts. The variable parts of a template can be filled in by other syntactic trees, which themselves are recursively generated from syntactic templates, and ultimately from syntactic templates without variable parts. A simplified example template is given in figure 2. The system currently contains approximately 30 templates for full sentences and approximately 60 templates for filling variable parts. In many cases, the latter contain variable parts themselves as well.

Of course, sentence generation by means of surface syntactic templates has its limitations. However, surface syntactic templates could be replaced by more abstract syntactic representations (e.g. syntactic derivation trees (cf. (Rosetta 1994)) or even semantic representations. A sentence generator of the kind used in ROSETTA (Rosetta 1994) or in PENMAN (Penman Natural Language Group 1989) would then have to be used to generate the actual surface strings. We emphasize that syntactic templates differ from what are called 'templates' in certain other approaches to natural-language generation (see e.g. (Uszkoreit (ed.) 1996)) in that syntactic templates are usually restricted to a single sentence, and are associated with a syntactic structure; they thus differ from templates which fix a larger piece of text and allow variability only sparsely, and from *schemas* (McKeown 1985), which are more abstract descriptions of the structure of a paragraph.

In the template of figure 2 a simplified syntactic structure is represented, and some variable parts are indicated. For convenience, we will not specify the syntactic structure of a template any further. Variable parts in a template will be represented between angled brackets. If we also have templates of the form *K. <number>*,[3] *Mozart*, *in <month>* and expressions for dates and numbers, it is possible to generate a sentence such as *K. 32 was written by Mozart in March 1772.* In this way it is possible to generate individual sentences. We will introduce further properties of templates relevant for generating coherent discourses below, where appropriate.

[3] Each composition by Mozart has been assigned a unique number, called its *Köchel-Verzeichnis*, abbreviated in English to *K.* plus a number.

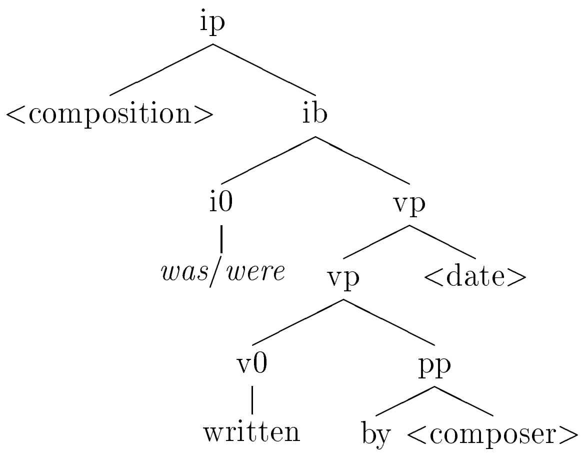

Figure 2: Template for a sentence such as ”K. 32 was written by Mozart in March 1772”. *Composition, composer, date* are the variables of the template. *Was/were* indicates that a choice must be made, depending on the subject. The labels *vp, v0* and *pp* stand for *verb phrase, verb* and *prepositional phrase*, resp. It is assumed that sentences (*ip*) are headed by an abstract *inflection* node (*i0*), which can contain auxiliary verbs and shows the normal projections in accordance with X-bar Theory, projecting $\overline{i}=ib$ and $\overline{\overline{i}}=ip$.

3.2 Generating discourses

Granted that sentences are available, how are coherent monologues generated? First, it has to be determined how the relevant information is to be presented. Various options are open to the user, from long monologues to no monologues (music only) and several intermediate options. These options correspond to different fillings of a variable called WHATTOTELL of the Dialogue state. This variable contains the entities, relations and attributes of the database which must be conveyed by the monologue generated. Second, it has to be determined about which composition or recording information is to be presented.This is represented by a specification of the selected recording and the selected recording part in variables of the Dialogue state called COMP and PART, respectively. Once this has been done, the language generation module is in a position to start generating the monologue.

An important characteristic of the language generation module in DYD is that the order of sentences and paragraphs is left free: it is not predetermined in the form of a grammar or schema or in any other manner. One could say that there is no explicit text grammar. (See (Odijk 1995a) for a description of this generation method from a different point of view.) Instead, each template, so to speak, ‘checks’ at any moment whether it is applicable, so that it can generate a sentence and have that sentence form part of the text generated. Whether a template ‘succeeds’ in doing so, depends on a number of properties associated with it, and on the current context.

Imagine that the language-generation module is somewhere in the middle of the process of generating a coherent text: Some sentences have been generated before, and some will be generated afterwards. At the same time, a partial context has been constructed and can be accessed in generating the current sentence.

In order to guarantee that the information conveyed is presented in natural groupings, a set of possible *topics* has been defined. Examples of such topics are *tells-about-composition, tells-about-recording, tells-about-performers, etc.*; their intended meaning is presumably clear from their names. The first thing the language module does is the following. It copies the value of the set of possible topics to a variable of the Topic state called TODOTOPICS. Next, it selects and removes an arbitrary topic from this set and makes it the CURRENTTOPIC. In addition, the Context state variables ATTEMPTEDTOPICS and DONETOPICS are initialized to the empty set.

Each template continuously 'checks' whether it can be applied, so that it can generate a sentence which fits in with the current context. What happens if not a single template can generate a sentence in the current context? There are two possible situations. Either not a single template was able to generate a sentence for the current topic earlier, or some templates were able to do so earlier but have changed the context in such a manner that no further templates are applicable. In the first case the current topic is added to the variable ATTEMPTEDTOPICS, a new current topic is selected and removed from the variable TODOTOPICS, and each template attempts to generate sentences again in the now updated context. In this situation the system has attempted to tell about the relevant topic 'too early', and it will try again later. In the second case the system has already conveyed the information concerning this topic and should start to convey information concerning a new topic. Therefore, a paragraph boundary is appended to the current monologue, and the Topic state is updated: the current topic is added to the variable DONETOPICS, the set ATTEMPTEDTOPICS is added to the TODOTOPICS, a new CURRENTTOPIC is selected and removed from the latter set, and each template starts checking again whether it can be applied in the now modified Topic state. Thus each paragraph of the monologue generated is associated with one *topic*, and the information is presented in a natural grouping. Information which logically belongs together is presented together and not scattered throughout the text. The process described continues until the set TODOTOPICS is empty, after which the generation system stops.

What conditions must be satisfied for a template to be applicable and able to get a sentence generated from it incorporated into the monologue? There are two types of conditions. A first set of conditions determines whether the template can be used at all in the current context. We will formulate these requirements in section 3.3.

A second set of requirements has to do with the sentences generated by the template. Each template generates a set of *candidate sentences*. If the template is to lead to a sentence appropriate in the given context, at least one of these candidate sentences must satisfy conditions with regard to (1) sentence-internal syntactic and semantic conditions on the use of referential and quantificational expressions, and (2) formal discourse-syntactic and discourse-semantic conditions on the proper use of referential and quantificational expressions. In evaluating the sentence-internal conditions context plays no role, so that we will leave these conditions undiscussed here. The discourse-syntactic and -semantic conditions, however, make essential use of a special part of the Context state and will be discussed in section 3.4.

3.3 When is a template applicable?

The conditions described in this section apply to the template and are checked before a single sentence is generated on the basis of the template.

The first requirement that a template must satisfy to be applicable is that it express the appropriate information, i.e. information which is contained in the Dialogue state variable WHATTOTELL, which gets its value when the user expresses, during the dialogue, how elaborate he or she likes the monologues to be. To get a grip on what information a sentences expresses, none of the usual formal semantic representations are used. Instead, each template is simply associated with a variable called TELLSABOUT which specifies which entities and attributes from the database the template expresses.[4] The relevant condition can be formulated simply as TELLSABOUT $\subseteq$ WHATTOTELL. This condition is essential to give the user control over what information the system will convey.

The second requirement that a template must satisfy in order to be applicable is that the information which it presents as 'new' information[5] has not been conveyed earlier in the text. This is to make sure that the system does not keep on repeating the same information over and over again. However, it is not possible to say that information which has been presented previously by the system should not be conveyed again. In fact, without this possibility it is not possible to obtain a coherent text at all. To appreciate, at an intuitive level, what is meant by presenting information as 'new' information, compare the following two examples:

(1) a The following composition is K. 32, Galimathias Musicum
b Mozart wrote this composition when he was only ten years old
c When Mozart was only ten years old, he wrote this composition

The discourse consisting of sentence (1a) followed by sentence (1b) is perfectly coherent and natural. That consisting of sentence (1a) followed by (1c), however, is not. But sentences (1b) and (1c) convey the same information, so the difference between the two discourses must be caused by some other factor. In the (b) sentence the information about the age at which Mozart wrote the relevant composition is presented as 'new' information. In sentence (c), however, the fact that Mozart *wrote* this composition is presented as 'new' information, but this is not appropriate in the current context. This difference is explicitly represented in the templates for (b) and (c).

To enable the requirement described above to be satisfied, various mechanisms have been added to the Context Model. One of the mechanisms which plays a role here will be discussed in section 3.4. A second mechanism, the Knowledge state, records for each attribute, relation and entity from the database (1) whether it has already or not yet been conveyed, (2) when it was conveyed, and (3) how it was conveyed.

The specification of *whether* some piece of information has already been conveyed is essential

[4] Given an interpretation of what these entities and attributes are, the information in WHATTOTELL, together with information about the referents of the fillers of the gaps in the template (see section 3.4), the truth conditions of a sentence generated on the basis of the template are, of course, easily determined.

[5] 'New' has been put between quotes since this notion is only to be made precise in section 4.

to guarantee that the system will not keep repeating the information and, if it is repeated, to guarantee that it is not presented as 'new' information .

The specification of *when* some piece of information has been conveyed is measured in our application in terms of a quadruple <presentation, text, paragraph, sentence>, indicating in which sentence of which paragraph of which text of which presentation the relevant information was conveyed. Below we will see that even the position within the sentence is recorded since this can be important in certain cases. For the moment, however, it is especially important that the Knowledge state records in which presentation a certain piece of information has been conveyed. This is important because the user can browse freely and end up selecting the same recording part, or a different part of the same recording as in one of the preceding presentations. The system takes this into account by keeping track of which recordings and which information about them have been presented, and when this has happened. The monologue adapts itself in various different ways to the state of the context. We will give some examples. If the recording selected was selected very recently before , the monologue will point this out, and not repeat all the information. If the recording selected was selected earlier, but not so recently, the monologue will point out that the recording was selected earlier and give a short summary of the major properties of the recording. It is useful to keep track of previous selections and the information conveyed about them in the Context Model even if the current selection was not selected earlier. In this situation the system can compare the current selection with the previous selections and the monologue can point out significant differences and/or similarities between the current selection and the previous ones.

The specification of *how* a particular piece of information has been conveyed distinguishes two cases: the information was presented *explicitly* or *implicitly*. To illustrate what is meant by this distinction we can cite the example of expressing dates. If the date of a certain event is known exactly (e.g. 10/8/1767), it can be expressed *explicitly*, for example with the expression *the tenth of August 1767*. But there are still other ways of expressing it more *implicitly*. Two subcases can be distinguished here. First, a date can be stated very precisely, but in such a way that its exact extension can be determined only if the user has sufficient relevant knowledge, e.g. by an expression such as *when Mozart was only twelve years old* or (for another date) *two days before his 35th birthday*, etc. The information about Mozart's age is not directly represented in the database, but must be computed on the basis of his birth date and the date to be expressed, which are available in the database. Second, the date can be expressed in a less precise manner, e.g. *in the early seventies* for 6/4/1972, etc. In the current application it is often preferable to express dates and other detailed information either in incomplete detail or indirectly. The language-generation module can generate such expressions, and, since the database contains a list of templates conveying events from Mozart's biography, it can also relate concrete dates to such templates and express an event from Mozart's biography, e.g. *when the Mozart family visited the Netherlands* instead. In all these examples the relevant date is given only implicitly and this fact should be recorded in the Knowledge state: if this is recorded, it is possible to present the exact date as 'new' information if the user happens to be interested in it, or if this is necessary in order to make appropriate comparisons with recordings presented later. The context must therefore be extended by adding the history of recordings presented, and a specification of what information has been conveyed, when it has been conveyed, and how it has been conveyed.

We now turn to the third requirement that a template must satisfy, which concerns *topics*. Each template is associated with one or more topics. The CURRENTTOPIC from the Topic state must be one of the topics associated with the template, otherwise the template cannot be used in the current context. This is a necessary ingredient of the system since it guarantees that information which logically belongs together is presented together and not scattered throughout the text.

The fourth requirement a template has to satisfy is that it is stylistically appropriate. This aspect has not yet been fully implemented in the system. What we mean by it is that it should be ensured that the same construction or the same words are not used too often too close together. If a template satisfies all other requirements mentioned above but has been used very recently or contains lexical items which have been used very recently, the system should not use this template if there are alternative templates which can be used to express the same information. We do not have any experience with such a mechanism and many questions are still open (e.g. when are two templates 'too close together', what does 'too often' mean), but such a mechanism is necessary in some form and it requires that the Context state keeps track of which templates and lexical items have been used before and when they have been used.

In this section we have discussed four conditions that templates must satisfy to be applicable:

(1) The template must present the information to be conveyed,
(2) Only new information must be presented as new information,
(3) The template must express the right topic, and
(4) The template must be stylistically appropriate.

In the next section we will discuss various conditions which the sentences generated from templates must meet.

3.4 Discourse-syntactic and -semantic conditions on sentences

If a template satisfies all the requirements described above, it can be used in the current context to generate sentences. If more than one template satisfies all the requirements described above, one is arbitrarily selected. In the DYD system the selected template first generates a set of *candidate sentences*. These candidate sentences are alternative ways of expressing the same information but their appropriateness depends on whether the referential and quantificational expressions used in the candidate sentence are compatible with the current Context state. The information in the context that is particularly relevant here has been represented in a separate part of the Context state and is called the *D-model* (for *discourse model*). Special conditions have been formulated which check whether a candidate sentence can be used given the current state of the D-model. If more than one candidate sentence satisfies all the conditions, one is chosen randomly to be actually used in the monologue being generated.

We will first give an illustration of some of the conditions we have in mind, and then we will describe in more detail how a D-model is constructed and what kind of information the D-model contains.

Many entities in the DYD domain (e.g. persons, compositions) are associated with a *proper name*, and they can be referred to by using this proper name. It is inappropriate to use two different proper names when referring to the same object, unless it can be presupposed that these two different names are known to refer to the same object. This can be illustrated with the following example. In the (mini-)discourse (2a) one may get the impression that *Eine Kleine Nachtmusik* and *K. 525* refer to two different objects. The discourse (2b), however, sounds completely natural.

(2) a ?You are now going to listen to K. 525. Eine kleine Nachtmusik is a sonata.
b You are now going to listen to K. 525. This composition, which is also known as 'Eine kleine Nachtmusik', is a sonata

In example (2b), the phrase *Eine kleine Nachtmusik* either does not refer at all, since it is used predicatively, or it refers to the title of K. 525, not to the composition K. 525. Therefore, this example does not violate the condition mentioned above.

Use of the same proper name over and over again can be avoided in various ways, e.g. by using pronouns such as *he, it, they* instead. But pronouns are also subject to various conditions relating to the context. For instance, they must have an antecedent in the current sentence or in the preceding discourse and the antecedent must not be 'too far away'. In the following example, discourse (3a) is ill-formed, since the pronoun does not have an antecedent in the preceding discourse. Discourse (3b), however, is perfect. Both the proper name and the pronoun are used appropriately:

(3) a *You are now going to listen to it. Eine Kleine Nachtmusik is a sonata.
b You are now going to listen to Eine Kleine nachtmusik. It is a sonata.

Various other expressions can be used to fill the gap in a template, each appropriate under specific conditions and imposing its own requirements on the context. Apart from proper names and pronouns, there are definite descriptions (basically, but somewhat simplified, NPs introduced by the definite article *the*, e.g., *the sonata, the composer*), demonstrative descriptions (NPs introduced by *this, these, that, those*, e.g *this composition*), indefinite descriptions (NPs introduced by the article *a*, e.g., *a sonata*), quantificational expressions, e.g. universally quantifying expressions (e.g. *each part, all compositions*), and probably many others. Universally quantifying expressions can arise through *aggregation*. Consider the following example. Certain recordings consist of a large number of parts (e.g. 12) which are all of the same kind (e.g., they are all *German dances*). It would be very unnatural if the monologue were to describe this by stating the kind for each individual part separately: *the first part is a German dance. The second part is (also) a german dance....The twelfth part is also a German dance.* Instead, it would be much more natural to say: *Each part is a German dance.* But the universally quantifying expression *each part* occurs in this sentence, and there are special discourse conditions applying to universally quantifying expressions.

As was explained in the Introduction, our starting point for the D-models were the discourse representation structures (DRSs) from Kamp's Discourse Representation Theory (DRT), but there are many differences between D-models in the DYD system and DRSs. For example, discourse-*syntactic* aspects play hardly any role in DRSs. A well-known example of a discourse-syntactic phenomenon relates to grammatical gender, which exerts its influence across sentences and cannot be reduced to existing semantic categories. (Odijk 1995b) argues that many other purely syntactic properties can also exert their influence across sentential boundaries, e.g. whether a sentence is headed by an infinitive, a participle or a finite verb, whether an infinitive is accompanied by *to* or not, whether a sentence is introduced by an overt complementizer or not, etc. For this reason, the D-model consists of both a syntactic part and of a semantic part. The syntactic part contains the syntactic information of expressions which is relevant for discourse syntactic conditions.

In addition, there was no need in the specific application to compute a full semantic representation, as is done in DRSs. The D-model therefore does not contain a full semantic representation of the preceding discourse but only a semantic representation of certain parts of it, and only to the extent that these are necessary to enable the system to deal adequately with the various referential and quantificational expressions.

Furthermore, the way DRSs are constructed is extremely complex. Here is an example: in a sentence such as (4a), first a part of the main clause must be processed (the part containing *K. 32*), then the subordinate clause, and finally the rest of the main clause. In (4b), however, a different part of the main clause must be processed first (a part containing *Mozart*), then the subordinate clause, and then the remaining part of the main clause. It is also easy to construct different sentences which must be processed in still other orders.

(4) a Since Mozart wrote it at a very early age, K. 32 made him instantly famous all over Europe

b Since he wrote K. 32 at a very early age, it made Mozart instantly famous all over Europe

The only way to do so is to make the procedure highly non-deterministic. Incorrect paths will be eliminated automatically. Non-determinacy is no problem from a purely theoretical point of view, but in a practical application more efficient deterministic procedures would be preferable wherever possible. Finally, allowing such nondeterminism does not work for certain definite descriptions without additional constraints (see (Odijk 1996)).

Another reason why the construction of DRSs is so complex is that DRT also attempts to account for various sentence-internal syntactic conditions on the use of various expressions in the algorithm for constructing DRSs. In DYD, we kept these sentence-internal conditions separated, and implemented a (currently simplified) version of the Binding Theory (see (Chomsky 1981), (Chomsky 1986) and, for a variant in the HPSG framework, (Pollard and Sag 1994)), which applies before the construction of the D-model. We think that a combination of Binding Theory (for sentence-internal syntactic conditions) with discourse conditions on referential expressions is to be preferred to an approach where both are dealt with by a single mechanism. (See also (Asher and Wada 1988) for some relevant discussion.)

D-models are constructed in a completely deterministic manner. The order in which the various elements are added to the D-model is fixed in a particular way but can be chosen arbitrarily. This is possible because the relevant conditions are checked only after the current D-model is extended to include the relevant information from the current sentence, i.e., a candidate D-model is constructed first, after which well-formedness conditions applying to D-models check whether the D-model is well-formed. If not, the corresponding sentence cannot be used in the current context and a different candidate sentence must be checked for its appropriateness in the current context. This does require the type of the phrase (*definite description, pronoun, proper name, etc.* See the list above) to be encoded in the D-model.

The D-model contains syntactic information for each referential or quantificational expression such as grammatical gender, grammatical person, grammatical number, the type of the phrase, and the position of the phrase in the sentence. Also, the position of the sentence in which the relevant expression occurs is stored with each phrase. This position is obtained from a Context state variable LOCATION, which keeps track of which sentence of which paragraph of which text of which presentation the system is in.

The D-model also contains semantic information for each phrase to the extent that this is necessary in order to guarantee the proper use of referring and quantificational expressions. Examples of semantic information are the natural gender of a phrase, a semantic representation of a phrase (especially necessary for definite and demonstrative expressions), and a semantic sorting of a phrase. The latter is required since there are restrictions applying to anaphoric expressions and their antecedents which can be formulated in terms of 'degree of specificity'. The degree of specificity of an expression is determined by the place of its associated sort in a sort hierarchy.

Various conditions as to the proper use of the various types of expressions can now be formulated as well-formedness conditions on D-models. One such condition states that a pronoun requires an antecedent which must occur in the same sentence or in the immediately preceding sentence (Pinkal 1986:370). This condition can be expressed by making use of the information in the D-model about the positions of the sentences in which a candidate antecedent and the pronoun occur. Initial versions of conditions of this kind have been formulated in the current system. Future versions are foreseen to profit from some insights stemming from *Centering Theory* (e.g. (Grosz *et al.* 1995)).[6]

4 Context in the generation of prosodic structure

In practical applications involving the generation of spoken language, the quality of the speech output can be an important variable determining the success of the product, and this quality, in turn, depends in large part on the 'melodic' (i.e., nonsegmental) properties

[6] Adding centering rules to DYD seems feasible. For example, the 'backward-looking center' (Cb) of a sentence (i.e., roughly, its main anaphoric element) can be determined apart from context (e.g., by identifying it with the grammatical subject of the sentence (Gordon *et al.* 1993)). Consequently, the Cb can easily be encoded in the templates. A similar observation holds for the 'forward-looking centers' (Cf) of a sentence (i.e., the entities that are made available for further reference by the sentence), which can be simply read off the D-model. Presumably, the ranking (i.e., the prominence order) of the different elements of the Cf can be determined on the basis of the surface order of the words, plus the syntactic information in the template.

of the speech produced. For example, if the pitch height of an utterance is constant or declines only gradually, then the utterance is perceived as monotonous, boring, and difficult to understand. Of course, if large speech fragments are stored in their entirety, the problem of melodic quality can be finessed, but this is impossible in any system that requires spoken output of great variety, as is the case with the DYD system. But, *generating* speech with high melodic quality is still very difficult, no matter whether the utterances are generated from type-written text (as in 'text-to-speech' systems) or from semantic representations of some sort (as in 'concept-to-speech' systems). The key to this problem lies in determining the abstract, 'prosodic' structure of the utterances generated, which determines the linguistically relevant aspects of phrasing and accenting, abstracting away from details of acoustic realization. Commercially available speech-synthesis systems such as DECTALK cannot confer good prosodic structure on unannotated strings of words because of the problems involved in parsing and understanding such strings. On the other hand, these systems do typically allow strings to be annotated with indications for accents, prosodic boundaries (e.g., pauses), and the like. In other words, the generation of synthetic speech of high prosodic quality requires the texts generated to be enriched with linguistic information of the kind necessary to derive the required annotations. Such 'embedding' systems are now beginning to emerge and DYD is an example of this trend.[7] The present discussion focuses on accenting, since this is the aspect of prosody in which context plays by far the most prominent role.

It is probably fair to say that most speech-generation systems count on generating a plethora of accents. Speaking very roughly, they accent all content words except those whose roots are identical to the root of a word that has recently occurred (Hirschberg 1990). It is important to note, in the setting of the present paper, that even this simple approach makes a very essential use of the linguistic context of a word, modeled as a list of word roots. Not only is this strategy easy to implement but it also reflects a degree of experimental wisdom. For example, Terken and Nooteboom found that additional accents will seldom cause an utterance to become unacceptable (Terken and Nooteboom 1987). In other words, the lack of an accent carries more information than the presence of one, so generating a multitude of accents minimizes the risk of outright errors. On the other hand, this 'overkill' strategy is also thought to lead to monotonic speech (e.g., (Hirschberg 1990)). Moreover, an unmotivated accent can change meaning. For example, in

(5) a The composer of this piece was a creative person.
b $^{-}$Haydn was very old though.

(6) a The composer of this piece was a creative person.
b $^{+}$Haydn was very old though.

adding an accent to *Haydn* also changes the (b) utterance from one about the composer of this piece to one about some other individual.[8] For these two reasons, we have opted for a

[7] Another example is the Synphonics system which is currently being developed at Hamburg University. Like DYD, Synphonics seeks to avoid the generation of unnecessary accents (see below), but Synphonics does not employ a deaccenting mechanism of the kind used in DYD (Abb *et al.* 1996).

[8] In example sentences, a +/− marking preceding a word denotes the presence/absence of an accent on that word. The absence of a +/− mark on a word leaves it unspecified whether the word is accented or not.

strategy that differs sharply from the one just sketched. What we do is try to generate all and only those accents that are necessary, given the context and the intended interpretation of the utterance. The approach adopted to attain this goal starts from the proposals in (van Deemter 1994b), which, in turn, are an extension of the proposals in (Dirksen 1992).

What we will try to do in the present section is to sketch the just-indicated approach to accenting and show that the approach to language generation described in section 3 lends itself very well to the implementation of the proposals of (van Deemter 1994b) (see section 4.1). In particular, the Context Model that proved to be so central for the generation for coherent monologues will also prove to be central to the attribution of accents. The resulting system combines properties of concept-to-speech generation and phonetics-to-speech generation. On the one hand, the generation module always has a semantic representation of the facts expressed by a certain sentence. This resembles the situation in a concept-to-speech system and it allows generation to take semantic factors into account. On the other hand, each sentence is generated from a syntactic template, and this template may be enriched with several prosodic markers. This is similar to the situation in a phonetics-to-speech system, and it allows certain accents as well as phrase boundaries to be coded by hand. Moreover, the templates can be designed so as to contain whatever syntactic information may be needed by the prosodic module. For these reasons the DYD project and its approach to generation make DYD a suitable carrier for the implementation and testing of prosodic theories.

The following section will outline a theory of accenting and subsequent sections will show how this theory has been implemented in the DYD system.

4.1 A simple model of accenting

The problem of accenting as viewed in the DYD project is to make sure that accents are located 'at the right places' in the monologues generated and that they are acoustically realized in appropriate ways. Following the IPO school of intonation ('t Hart *et al.* 1990), we have assumed that the problem of accent *realization* belongs to the domain of the Speech module. The *Prosody* module makes no distinction between different kinds of accents (as is done, for example, in the Pierrehumbert school of intonation (Pierrehumbert and Hirschberg 1990)) and tells the *Speech* module where the accents are to be located rather than how they are to be realized in sound. The issue of how accents are realized is very briefly discussed in section 4.2. The rest of this section deals with the problem of accent location.

The 'right places' at which accents should be located are understood to be those in which accents would appear if the monologues occurred spontaneously. Note, however, that this could make accent location an incompletely defined problem since it has often been claimed that a monologue can typically be accented in more than one way (Terken and Nooteboom 1987). There are two ways of looking at this situation. One is to go along with the received view and introduce an element of nondeterminism. Our own perspective is a different one, in which we try to minimize nondeterminism. Thus, our working assumption will be that there must always be a reason why accents occur (or do not occur), and that a speech synthesis system improves as it gets a better grip on more and more of these 'reasons'.

The first step towards eliminating nondeterminism is to add meaning as a parameter and to require accents to be located at 'the right places' *given* the intended meaning of the monologue. Thus, both (5)(b) and (6)(b) are acceptable, but if *Haydn* and *The composer of this piece* are intended to be different individuals, then only (5)(b) is correct. What this means is that the context modeling required for accenting can no longer be a mere list of words and that it must contain information about the things these words refer to, i.e. with their semantics.

However, it seems unlikely that this move alone can eliminate all nondeterminism. Assume, for example, that *Haydn* and *The composer of this piece* designate the same person (i.e., they corefer), while Haydn is also contrasted with some other person, namely Mozart:

(7) a The composer of this piece was a creative person.
    b $^{+}$Haydn was very old, but $^{+}$Mozart was not.

(Note that this makes (7) ambiguous between a situation in which Haydn and the composer of this piece are the same person and one in which they are not.) This example shows that other factors than just novelty of information must be taken into account as accent triggers. Generalizing this idea, we arrive at a model of accenting in which various factors (novelty of information, contrast, etc.) can 'write' a marking on a 'blackboard' that will cause an accent to show up somewhere in the text. (The precise location of each accent is determined by considerations which will be discussed below.) In accordance with this hypothesis, DYD does not produce any accents other than those necessitated by one of the factors that write on the blackboard.

The remainder of this section will be devoted to an explanation of the notions of givenness and novelty of information, since it is vital to define rigorously what these amount to. A brief discussion of how other accent-triggering factors, such as contrast, are treated, will be postponed until section 4.2.

A proposal for definitions of givenness and newness was put forward in (van Deemter 1994b). The key concept in this proposal is that of *identity anaphora*, which stems from (van Deemter 1992). The idea is the following. Much of what is said in a discourse can be interpreted in two ways, namely as true of the entire domain of discourse or as true of some contextually determined subdomain. This is also true for the information contained in a noun phrase. For example, consider

(8) a The children were upstairs.
    b The girls were having fun.

Here the descriptive information contained in the noun phrase *The girls* (which says, of a certain set, that it contains all the girls that exist in a certain domain) may be true relative to the entire domain of discourse, but it may also be true relative to the domain that consists of just the earlier-mentioned children. In the latter case, it says that the set contains all the girls who are *also* elements of this set of children, and the predicate *were having fun* in (8) asserts that all of these (i.e., the elements of Girls $\cap$ Children) were having fun.

With regard to this second, contextual interpretation, two possibilities are distinguished: either all the children happen to be girls or some of them are boys. In the first case, we speak of identity anaphora, since the noun phrase *the girls* refers to an already familiar set of individuals. In the second case, where a subset of the set of children is carved out, we speak of subsectional anaphora. Both types of anaphora can be applied to a wide range of expressions, including both definite (pronominal or other) and indefinite noun phrases. It was hypothesized that the same ideas apply to VPs and other major phrases but this was not worked out in detail.

Later work has proposed to make use of this idea for the prediction of accents in discourse (van Deemter 1994b). The proposal boils down to the following:

> **Informational givenness of NPs.** If an NP can be construed as standing in a relation of identity anaphora to another NP, then it constitutes given information. If not, it constitutes new information.

One NP 'can be construed as' standing in a relation of identity anaphora to another if analyzing it as identity-anaphoric to this particular antecedent leads to the intended interpretation. If an NP can be construed as having identity anaphora to another NP, it will be called 'object-given'. NPs that constitute new information have to be accented, while NPs constituting given information may or may not be accented. To illustrate this let us assume that (9) is the first sentence of a text, so that initially everything is new information.

(9) a $^{+}$Haydn was a funny character.
b $^{-}$Haydn was really creative though.

Observe that, on a traditional analysis, (9) does not contain any anaphors. Yet, the rule implies that the second occurrence of *Haydn* must be deacccented, since it can be construed as an identity anaphor. Analyzing it as an identity anaphor means that the (second) NP *Haydn* picks out all those individuals in the singleton {Haydn} that are identical to the intended referent of the second occurrence of *Haydn*, which is another way of saying that the two occurrences of *Haydn* corefer. If the two do not corefer, the theory predicts that both occurrences are accented. Similarly, the theory predicts that *pronouns* are never accented for 'novelty of information' reasons, except when the descriptive information in the pronoun is used to act as a subsectional anaphor.[9] For specifics and formal mechanism the reader is referred to (van Deemter 1992) and (van Deemter 1994b). In section 4.2 we shall see that the theory just outlined was simplified considerably in the implementation of DYD.

For now, let us assume that it is known of each major phrase whether it must be accented or not. If a constituent is accented, Focus-Accent theory offers rules that determine what word in the constituent the accent must 'trickle down to'. For example, if the phrase *the author of a sonata* is used to introduce a new individual into the discourse, it needs to be accented. Focus-Accent theory predicts that, normally, the accent will land on *sonata*. However, there can be several reasons why accent is prevented from going to that part of the syntax tree, and these are covered by the so-called *Default Accent* Rule (Ladd 1980). In (van Deemter 1994b) it was proposed that deaccenting of a word can take place for several reasons. One reason is that the word is part of an NP that has identity anaphora to some other NP, as in (10), where it is assumed that *this sonata* refers to *this piece of music*:

(10) Look at this piece of music. [The $^{+}$composer of [this $^{-}$sonata]] must have been a funny guy!

Let us assume that the subject of the second sentence designates an individual who has not been mentioned in the monologue so far, and therefore contains new information. Since *this sonata* is identity anaphoric to some already established discourse entity, the Default Accent

[9] For example, some speakers find the following sequence of sentences acceptable, but only if *She* is accented: *The married couple came out of the City Hall.* $^{+}$She *looked gorgeous.*

Rule will move the accent to *composer*. Note that this happens even if the word *sonata* itself has not been mentioned before. The second reason for deaccenting occurs, roughly, when the word that would normally receive accent has occurred in the very recent past, or when a subsumed (i.e., extensionally included) word has done so. Note that this makes concept-givenness a nonsymmetrical relation, since a word may now be deaccented because of a subsumed word, but not the other way round. For example, in (11), *string instruments* can be deaccented because of the extensionally included *viola.* In (11), however, *viola* cannot be deaccented because of the subsuming (i.e., extensionally including) *string instruments.*

(11) a Bach wrote many pieces for $^+$viola; he must have loved $^-$string $^-$instruments.
b Bach wrote many pieces for $^+$string instruments; he must have loved the $^+$viola.

So much for the treatment of accents stemming from novelty of information. A more detailed explanation, which will also contain a modest account of some other types of accents (e.g., accents stemming from contrastivity of information) follows in section 4.2 in connection with the DYD system.

4.2 Implementation of the model for accenting

In the present section the implementation in DYD of the earlier-described model of accenting will be discussed. The discussion will focus on considerations of givenness and novelty since these are most directly connected with the issue of context modeling. We will continue to concentrate on prosodic structure but the section will conclude with a brief sketch of the Speech module of DYD, which is responsible for the acoustic realization of prosodic structure.

As a starting point for DYD's *Prosody* module, we took Arthur Dirksen's implementation of Focus-Accent Theory ((Dirksen 1992)) and added semantic information to it. This implementation takes syntactic trees of a conventional kind as input and converts them to metrical trees, each of which is built up from binary and unary branching nodes whose nodes are marked as *strong* or *weak.* In English, the rightmost branch of a binary node is almost invariably marked as *strong*, while the leftmost branch is marked as *weak.* The only daughter of a unary node is always strong.

In our first implementation novelty of information was the only accent-triggering factor taken into account. Initially, all major phrases are marked $+F$. Now, if a given node is marked $+F$, then it is also marked as *accented*, and so is each strong node that is the daughter of a node marked as *accented.* In principle, accent materializes on those leaves that are marked as *accented.* However, there may be several obstacles preventing this from happening. Leaves may end up unaccented in several circumstances, as we have seen in the previous section. In particular, a node may be marked $-A$ ('accent is blocked') in any of the following cases:

Accent blocking:
(a) A major phrase is marked $-A$ if it is *object-given.*
(b) A major phrase is marked $-A$ if it is *always-given.*
(c) A leaf x is marked $-A$ if it is *concept-given.*

(d) A leaf is marked $-A$ if it is marked as *lexically unfit* to carry an accent which is due to novelty of information.

Clauses (a)-(c) deal with three different kinds of givenness, while (d) is a purely lexical affair. All of these will be explained below.

Object-givenness. (Clause a) We have seen in section 4.1 that an NP is object-given if it can be construed as having identity anaphora to another NP. Complex as this might seem, the way in which marking for object-givenness is implemented in DYD can be simple, thanks to the fact that DYD's Context Model contains all the relevant information. For NP_1 to be able to be construed as having identity anaphora to NP_2, two conditions suffice:

1. NP_1 and NP_2 corefer.
2. The structure of the monologue is such that NP_1 can have NP_2 as its antecedent.

The information in **(1)** is contained in the D-model and there is a rule which determines whether **(2)** holds given the information contained in the D-model.

Always-givenness. (Clause b) Deaccenting should not only be triggered by the referents of expressions that occur explicitly in the monologue. For example, objects can be *indirectly introduced* into the discourse model, as in

(12) This suite was first performed in Prague. It is said that the audience was enthralled by the performance.

where the performance is indirectly introduced into the context by the verb phrase *was first performed in Prague.* Such difficult cases are left for future research. DYD does implement another, simpler category of cases in which an object exists in the Context Model without being explicitly introduced. An example in the DYD domain is Mozart himself. By including this person in the context state, many distracting accents on the expressions *Mozart*, *Wolfgang Amadeus*, etc. are avoided. Thus, if the following sentence were to be uttered as the first sentence of the first monologue, the word *Mozart* would not be accented:

(13) Mozart composed the following composition when he stayed in Vienna.

Concept-givenness. (Clause c) The crucial notion for concept-givenness is that of subsumption of a recent expression. The key concepts of recency and subsumption are implemented as follows. 'Recent' is interpreted as 'to the left of the subsuming expression' and 'contained in the same or in the previous sentence'. The subsumption relation is implemented as follows:

Subsumption: x subsumes y if
(1) $x = y$;
(2) x and y are listed as synonyms; or
(3) $\langle x, y\rangle$ is an element of the transitive closure of the relation R.

The relation R is stipulated by means of a list of pairs of words.[10] For example,

$$\{\langle mozart, composer\rangle, \langle composer, man\rangle, \langle man, person\rangle\}.$$

The occurrence of a pair $\langle x, y\rangle$ in R implies that either the denotation of x is an element of that of y (as in $\langle mozart, composer\rangle$), or the denotation of x is a subset of that of y. For example, the pair $\langle composer, man\rangle$ indicates the fact that, in our database, all the composers happen to be men. In principle, this strategy is believed to give the desired results for words of all syntactic categories (for example, in the category of determiners, one could use such pairs as $\langle some, many\rangle$, $\langle many, all\rangle$, etc.), but the present implementation restricts itself to pairs of nouns.

Lexically unfit. (Clause d) Some words do not 'like' accent. Examples include the articles *the* and *a(n)*, as well as most prepositions (*in*, *on*, *about*) in most of their uses. By marking them as lexically unfit to carry accent, we exclude the possibility that accents related to novelty of information land on these words.[11]

A $-A$ marking causes the so-called Default Accent rule to be triggered, which transforms one metrical tree into another:

[10] More precisely, the pairs consist of words that are disambiguated by subscripts. For example, the word *second* has two versions, 'second$_1$' and 'second$_2$', one of which denotes a number and the other a unit of time measurement.

[11] Note that accents related to other factors, such as contrast, *can* land on these words, which is why lexical unfitness is not allowed to block accents related to those factors.

> **Default Accent Rule:** If a *strong* node n_1 is marked $-A$, while its *weak* sister n_2 is not, then the *strong/weak* labeling of the sisters is reversed: n_1 is now marked *weak* and n_2 is marked *strong.*

The Default Accent Rule may cause accent to move to a word where syntactic factors alone would never place it. To illustrate this, consider the following piece of discourse:

(14) a You have selected K.32.
 b You will now hear K.32.

In English it is usually the right daughter of a mother node that is *strong.* Thus the initial metrical tree looks as in figure 3. The Verb Phrase *will now hear K.32* is marked $+F$ and therefore labeled as *accented.* Now if semantic factors did not intervene, *K. 32* would carry an accent since the *accented* label would trickle down along all the rightmost branches of the tree. But since *K. 32* is also referred to in the previous sentence of the discourse, *K. 32* is object-given, and marked $-A$. As a result, the Default Accent rule swaps the *strong/weak* (S/W) labeling between *hear* and *K. 32* before the *accented* labels are assigned. Consequently, the sentence accent trickles down along a path of *strong* nodes and ends up on *hear.* Note that, in accordance with the discussion in section 4.1, clause (c) of the markings for Accent Blocking makes use of a nonsymmetric relation between anaphor and antecedent, requiring the 'antecedent' to be subsumed by the 'anaphor', rather than the other way round. The following accent pattern is predicted:

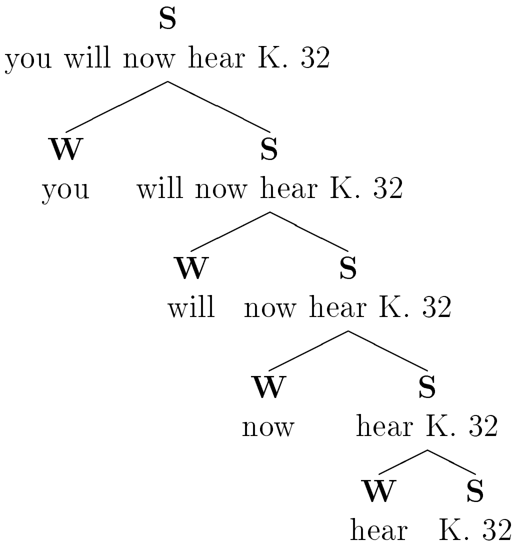

Figure 3: Example of a metrical tree

(15) a Two people heard the performance; $^{-}$Three $^{+}$composers had been invited.
 b Two composers heard the performance; $^{+}$Three $^{-}$people had been invited.

The reason is that, due to the asymmetry of clause (c), *people* in (b) is marked $-A$, while *composers* in (a) is not.

The treatment of novelty-related accents described in the present section leads to some promising results,[12] but an initial implementation showed some shortcomings, not the least of which was that novelty of information is the only accent trigger that was used. Thus, for example, there was nothing to guarantee an accent on expressions that are contrasted with other expressions, unless the items contrasted also happened to express new information. Thus, in a context where both *Mozart* and *Haydn* have recently occurred, these two names would not be accented:

(16) a $^{-}$Mozart wrote few string quartets.
b $^{-}$Haydn wrote many.

Of course, the accenting model allows contrast to be added as another source of accents, alongside novelty of information. Given the implementation of the accenting model sketched earlier in this section, this can be done by attributing a $+F$ label to all those expressions (including, but not restricted to, major phrases) that can be characterized as standing in contrast with some other expression. However, characterizing contrast is difficult. Arguably, determining the circumstances under which two expressions are in contrast requires an ability to recognize whether two sentences are in logical conflict with each other (van Deemter 1994a). In the case of DYD, however, the problem of contrastive accent may be finessed by exploiting the fact that generation in DYD makes use of templates. For example, consider the following template, where a hand-coded accent is indicated by a + following the constituent to be accented:

(17) ⟨This composition⟩ is (also | again) a ⟨ duo ⟩ in ⟨ G ⟩, like the
preceding composition, but now for ⟨ violin and viola ⟩+.

Since the addition of hand-coded accents is not specifically related to the issue of context modeling - even though the phenomenon of contrast is obviously a contextual one - we will not discuss in detail how hand-coding was used to improve system performance. In particular, we will refrain from discussing the circumstances under which a part of an expression marked as contrastive information can be subject to deaccenting. Instead, we will use the remaining part of this section to explain briefly how the prosodic structures that have been discussed are used to arrive at spoken output.

As was explained earlier on, the *Prosody* module of the system adds information about accenting and phrasing to the output of the language- generation module. The prosodically enriched sentences are then passed on to the *Speech* module (Fig.1), whose job it is to 'realize' this abstract structure in sound. An enriched sentence is transformed into a structure in which the abstract information concerning accenting is 'interpreted' in terms of Rises and Falls, in accordance with the IPO model of intonation ('t Hart *et al.* 1990, Willems *et al.* 1988). The resulting structure is then sent to the DECTALK speech-synthesis system, which takes care of the final phase of acoustic realization. As a result of this procedure, DYD

[12] For some preliminary experimental evaluation, see (van Donselaar 1995b, van Donselaar 1995a).

produces far fewer accents (roughly two-thirds) than it would do if sentences were sent directly to DECTALK, without prosodic enrichments stemming from the Context Model, resulting in less monotonic speech that is easier to interpret. For further explanation of the speech module, see (van Deemter *et al.* 1994). Examples of the accent patterns generated can be found in the Appendix.

5 Conclusions

We have described an actually implemented system that aims at the generation of coherent, correctly pronounced monologues on Mozart's instrumental compositions. The lack of a sound and generally accepted methodology for the evaluation of generation systems is notorious (Cole *et al.* 1995), which is why the present paper concludes with an Appendix in which the output of the system is exemplified.

We have shown that context modeling plays a vital role in various modules of this system, to guarantee that monologues

> **(1)** have a content that is appropriate given the wishes of the user, as expressed during the dialogue,
> **(2)** are linguistically coherent, both internally and in the larger setting of the interaction between system and user, and
> **(3)** are intonationally adequate, especially in terms of their accenting patterns.

We hope to have shown that a proper modeling of context is an absolute requirement if the type of system described is to function properly, and that the system developed forms an excellent tool for investigating the various issues at hand. In particular, the fact that DYD is a language-*generation* system implies that many things are under full control of the designer, so that variation can be introduced in exactly those aspects which require investigation. In addition, the design of the generation module, in which no standard discourse grammar is used, opens the way for various rules that safeguard the coherence of the monologues and make the role of context explicit.

We have also seen that 'context' encompasses various kinds of information. Some of these are purely syntactic in nature, others semantic; some are probably required in any speech-generation system, while others are highly dependent on the particular application. A so-called Context Model has been set up to keep track of all the relevant information, and rules have been formulated that ensure the fulfillment of the requirements covered by **(1)**-**(3)**. The Context Model can be thought of as a computationally viable version of a huge Discourse Representation Structure which is simplified in some (i.e., semantic) respects and enriched in other (mostly, form-related) respects. We will refrain from discussing the question of whether it is possible to embed our Context Models in a fully general theory of context - perhaps of the kind proposed by Buvač and Mason (Buvač and Mason to appear). Such questions require a very different, formally more sophisticated treatment and are left for further research.

6 Appendix

This Appendix contains an example of a possible dialogue between the system and a user, in which the user makes three successive requests, interleaved with three monologues on the part of the system. The texts are rendered without paragraph structure. Sentences are rendered verbatim and accents are indicated by means of + markings following each accented word. Accents in the dialogue part are omitted. The reader should bear in mind that not all the details of rules related to accenting have been explained in the present paper. In particular, this is true of hand-coded accents, but also of some rhythm-based rules as well as for some exceptions against the conditions under which constituents are deaccented.

What follows is the exact output produced by DYD as a result of the sequence of commands used in section 2 to exemplify the working of the dialogue module. First, the user request is repeated (in abbreviated form), then a brief response is shown in which the system reports back on what it has found, and then the monologue is presented. In this particular scenario, first (1), the user asks for a *sonata*, whereupon the system selects and presents a sonata for violin and harpsichord. Next (2), the user adds the condition that the composition has to contain a *piano*, whereupon the system selects a sonata for violin and piano. Finally (3), the user adds that it has to be a *fast* piece of music; the system interprets this as a request for compositions with tempo indication 'allegro' or 'presto' and selects one such composition. After each monologue a couple of shortcomings - as perceived by the authors - are briefly listed.

User request (1): *sonata.*
System response: *You asked for a sample in the sonata genre. I found 163 tracks. You will hear an example.*

> *The following+ composition+, from which you are going to hear a fragment+ of part one+, was written+ by Mozart in his Salzburg+ period, in Salzburg, in Brussels+, and in Paris+. The KV+ number of this work+ in c+ is K. six+. It is also+ known as the first+ sonata+ for violin+, and harpsichord+. The violin+ is played by Gerard Poulet+, and the harpsichord+ by Blandine Verlet+. The violin+ sonata consists of four+ parts+: allegro+, andante+, minuet tempo di menuetto+ - tempo di menuetto, and allegro molto+. This allegro is located on the first+ track+ of the second+ CD+ of volume fifteen+. This allegro lasts four+ minutes+ twenty+ nine+ seconds+. The recording+ of this sonata+ was made in Paris+, France+, in the seventies+. The quality+ of the recording of this work is ADD+. The following+ is a fragment+ of this allegro+.* (There follows a fragment.)

Shortcomings: (a) The *from which* construction is slightly awkward, but can easily be replaced. (b) The second occurrence of *this allegro* is infelicitous. (*It* would have been better.) (c) It seems clearly inappropriate that the second occurrence of *Salzburg* is deaccented. (d) Likewise, the second occurrence of *tempo di menuetto* could better be accented.

User request (2): *piano.*
System response: *You asked for a sample with pianos. I restricted your selection to 117*

tracks. You will hear an example, with one piano.

> *The following+ composition+, from which you are going to hear a fragment+ of part one+, was written+ by Mozart on the tenth+of July+ seventeen+ eighty+ eight+, in Vienna+. The KV+ number of this sonata+ in f+ is K. five+ four+ seven+. This composition is also+ known as the 36th+ sonata for violin+, and piano+. The recording+ was made in Dresden+, Germany+. The quality+ of the recording of K. five+ four+ seven+ is DDD+. The violin+ sonata consists of three+ parts+: andante cantabile+, allegro+, and andante. The first+ part+ is located on track one+ of the first CD+ of volume fifteen+. It lasts three+ minutes+ fifty+ eight+ seconds+. The following+ is a fragment+ of the first+ part+. The violin+ is played+ by Isabelle van Keulen+, and the piano+ by Ronald Brautigam+.* (There follows a fragment.)

Shortcomings: (a) The second occurrence of *K. five four seven* seems slightly verbose. (b) The second occurrence of *andante* (in *and andante*) should not have been deaccented.

User request (3): *fast.*
System response: *You asked for a sample with a high tempo. I restricted your selection to 72 tracks. You will hear an example, with tempo presto.*

> *The following+ composition+, from which you are going to hear a fragment+ of part three+ , was written+ by Mozart in the beginning+ of seventeen+ seventy+ five+, in Munich+. The work is also+ a sonata+ in f+, like the preceding+ composition, but now+ for piano+. The KV+ number of this work is K. two+ eight+ zero+. This sonata+ consists of three+ parts+: allegro assai+, adagio+, and presto+. The presto lasts two+ minutes+ forty+ five+ seconds+. This presto is located on track six+ of the first+ CD+ of volume seventeen+. The piano+ is played by Mitsuko Uchida+. The recording+of the sonata+ was made+ in the Henry Wood+ Hall in London+, England+, in the eighties+. The quality+ of its recording is DDD+. The following+ is a fragment+ of the third+ part+.* (A fragment follows.)

Shortcomings: (a) The template *also a .., but now ...* had better be rephrased. (b) *The presto* had better not be deaccented. (c) *This presto* might be replaced by a pronoun.

References

B. Abb *et al.* (1996), "Incremental grammatical encoding: an outline of the SYNPHONICS formulator," *Trends in Natural Language Generation: an Arificial Intelligence Perspective*, ed. by G. Adorni and M. Zock, (Springer Verlag, Berlin and New York)

N. Asher and H. Wada (1988), "A computational account of syntactic, semantic and discourse principles for anaphora resolution," *Journal of Semantics*, vol. 6, pp. 309–344

L. Augusteijn (1990), "The Elegant compiler generator system," *Attribute Grammars and their Applications*, ed. by P. Deransart and M. Jourdan, (Springer Lecture Notes in Computer Science, Berlin)

J. Barwise (1985), "Noun phrases, generalized quantifiers, and anaphora," Informal Note, CSLI, Stanford

S. Buvač and I. Mason (to appear), "Metamathematics of context," *Fundamenta Informaticae*, vol. 23, no. 3

D. Carter (1987), *Interpreting anaphors in natural language texts*, (Ellis Horwood, Wiley & Sons, New York)

N. Chomsky (1981), *Lectures on Government and Binding*, (Foris, Dordrecht)

N. Chomsky (1986), *Knowledge of language. Its Nature, Origin and Use*, Convergence, (Praeger, New York)

R. Cole *et al.* (1995), "The challenge of spoken language systems: Research directions for the nineties," *IEEE Transactions on Speech and Audio Processing*, vol. 3, no. 1, pp. 1–20

C. Condoravdi and M. Gawron (1996), "The context-dependency of implicit arguments," *Quantifiers, Deduction, and Context*, ed. by Kanazawa, Pinon, and D. Swart, (CSLI Publications, Stanford University)

K. van Deemter (1992), "Towards a generalization of anaphora," *Journal of Semantics*, vol. 9, pp. 27–51

K. van Deemter (1994a), "Contrastive stress, contrariety, and focus," *Focus and Natural Language Processing*, ed. by P. Bosch and R. v.d. Sandt, (IBM Working Papers No.6)

K. van Deemter (1994b), "What's new? A semantic perspective on sentence accent," *Journal of Semantics*, vol. 11, pp. 1–31

K. van Deemter, J. Landsbergen, R. Leermakers, and J. Odijk (1994), "Generation of spoken monologues by means of templates," *Proceedings of TWLT 8*, (Twente University, Twente), pp. 87–96, IPO MS. 1053

A. Dirksen (1992), "Accenting and deaccenting: a declarative approach," *Proc. of COLING*, Nantes, France

W. van Donselaar (1995a), "The function of de-accentuation in speech processing: A comparison of implicit and explicit references," Submitted to: *Language and Speech*

W. van Donselaar (1995b), "Listeners' use of the 'information-accentuation' interdependence in processing implicit and explicit references," To appear in: Proceedings Eurospeech '95, Madrid

P. Gordon, B. Grosz, and L. Gillion (1993), "Pronouns, names, and coherence," *Cognitive Science*, vol. 17, pp. 311–347

J. Groenendijk and M. Stokhof (1991), "Dynamic predicate logic," *Linguistics and Philosophy*, vol. 14, pp. 39–100

B. J. Grosz, A. K. Joshi, and S. Weinstein (1995), "Centering: A framework for modeling the local coherence of discourse," *Computational Linguistics*, vol. 21, no. 2, pp. 203–225

J. 't Hart, R. Collier, and A. Cohen (1990), *A Perceptual Study of Intonation*, (Cambridge University Press, Cambridge)

I. Heim (1982), *The semantics of definite and indefinite noun phrases*, Ph.D. thesis, University of Massachusetts, Amherst, Mass.

I. Heim (1992), "Presupposition projection and the semantics of attitude verbs," *Journal of Semantics*, vol. 9, pp. 183–221

E. Hinrichs (1986), "Temporal anaphora in discourses in English." *Linguistics and Philosophy*, vol. 9, pp. 63–82

J. Hirschberg (1990), "Accent and discourse context: assigning pitch accent in synthetic speech," *Proc. of AAAI*, p. 953

H. Kamp and U. Reyle (1993), *From Discourse to Logic*, vol. 42 of *Studies in Linguistics and Philosophy*, (Kluwer Academic Publishers, Dordrecht)

D. Kaplan (1979), "On the logic of demonstratives," *Contemporary Perspectives on the philosophy of language*, ed. by P. French, T. U. Jr., and H. Wettstein, (Univ. of Minnesota Press, Minneapolis), pp. 401–412

E. Klein (1980), "A semantics for positive and comparative adjectives," *Linguistics and Philosophy*, vol. 4

D. Ladd (1980), *The Structure of Intonational Meaning: Evidence from English*, (Indiana University Press, Bloomington, In.)

W. Mann (1987), "Rhetorical Structure Theory: description and construction of text structure," *Natural Language Generation*, (Martinus Nijhoff Publishers, Dordrecht)

K. R. McKeown (1985), *Text Generation. Using Discourse Strategies and Focus Constraints to Generate Natural Language Text*, Studies in Natural Language Processing, (Cambridge University Press, Cambridge)

R. Montague (1974), "Pragmatics," *Formal Philosophy*, ed. by R. Thomason, (Yale University Press, New Haven and London), pp. 95–119

H. Ney, V. Steinbiss, R. Haeb-Umbach, B.-H. Tran, and U. Essen (1994), "An overview of the Philips research system for large–vocabulary continuous–speech recognition," *Int. Journal of Pattern Recognition and Artificial Intelligence, Special Issue on Speech Recognition for Different Languages*

J. Odijk (1995a), "Generation of coherent monologues," *CLIN V: Proceedings of the Fifth CLIN Meeting*, ed. by T. Andernach, M. Moll, and A. Nijholt, University of Twente, Enschedé, pp. 123–131, IPO MS 1082

J. Odijk (1995b), "Topicalization of sentences in Dutch is contrastive dislocation (Dutch audience version)," IPO Report 1079, Institute for Perception Research (IPO), Eindhoven

J. Odijk (1996), "A syntactic condition on definite descriptions," *IPO Annual Progress Report 30 (1995)*, (Institute for Perception Research (IPO), Eindhoven), pp. 114–123

B. Partee (1989), "Binding implicit variables in quantified contexts," *Papers of the Chicago Linguistic Society, vol. 25*, pp. 342–365

Penman Natural Language Group (1989), "The Penman user guide," Technical report, Information Sciences Institute, Marina del Rey, CA

J. Pierrehumbert and J. Hirschberg (1990), "The meaning of intonational contours in the interpretation of discourse," *Intonations in Communication*, ed. by P. Cohen, J. Morgan, and M. Pollack, (MIT Press, Cambridge, Mass.), chap. 14, pp. 271–311

M. Pinkal (1986), "Definite noun phrases and the semantics of discourse," *Proceedings of COLING-86, Bonn, August, 25th - 29th, 1986*, (University of Bonn), pp. 368–373

C. Pollard and I. A. Sag (1994), *Head-Driven Phrase Structure Grammar*, Studies in Contemporary Linguistics, (Center for the Study of Language and Information/The University of Chicago Press, Stanford/Chicago and London)

M. Rosetta (1994), *Compositional Translation*, vol. 273 of *Kluwer International Series in Engineering and Computer Science (Natural Language Processing and Machine Translation)*, (Kluwer Academic Publishers, Dordrecht)

R. van der Sandt (1992), "Presupposition projection as anaphora resolution," *Journal of Semantics*, vol. 9, pp. 333–377

C. Sidner (1979), "Towards a computational theory of definite anaphora comprehension in English," TR-537, MIT, AI Lab

J. Terken and S. Nooteboom (1987), "Opposite effects of accentuation and deaccentuation on verification latencies for 'given' and 'new' information," *Language and Cognitive Processes*, vol. 2, no. 3/4, pp. 145–163

H. Uszkoreit (ed.) (1996), "Language generation," *Survey of the State of the Art in Human Language Technology*, ed. by R. A. Cole, chap. 4, pp. 161–169, URL: http://www.cse.ogi.edu/CSLU/HLTsurvey/

N. Willems, R. Collier, and J. 't Hart (1988), "A synthesis schema for British English intonation," *Journal of the Acoustical Society of America*, vol. 84, no. 4