Computing prosodic properties in a data-to-speech system

M. Theune* and **E. Klabbers*** and **J. Odijk** and **J.R. de Pijper**
IPO, Center for Research on User-System Interaction
P.O. Box 513
5600 MB Eindhoven
The Netherlands
theune@ipo.tue.nl

Abstract

We propose a set of rules for the computation of prosody which are implemented in an existing generic Data-to-Speech system. The rules make crucial use of both sentence-internal and sentence-external semantic and syntactic information provided by the system. In a Text-to-Speech system, this information would have to be obtained through text analysis, but in Data-to-Speech it is readily available, and its reliable and detailed character makes it possible to compute the prosodic properties of generated sentences in a sophisticated way. This in turn allows for a close control of prosodic realization, resulting in natural-sounding intonation.

1 Introduction

The central topic of this paper is the problem of computing the prosodic properties of sentences generated in Data-to-Speech systems (i.e., systems which present data in the form of a spoken monologue - sometimes also called 'Concept-to-Speech' systems). We propose a set of rules for the assignment of prosodic properties which take an explicit discourse model into account. In contrast to Text-to-Speech systems (and more generally, systems which require linguistic analysis of the input), explicit discourse models can be reliably constructed in Data-to-Speech systems (and more generally, in systems which generate natural language from data), so that a more natural prosody can be achieved.

The rules for prosody assignment described in this paper are used in the language generation component of D2S, a generic system for the creation of Data-to-Speech systems. The method for natural language generation implemented in D2S is hybrid in nature (Reiter, 1995); (Coch, 1996). It is a particular mixture of (syntactic) template-based techniques and full natural language generation, described in more detail in Klabbers et al. (1997a). A variety of Data-to-Speech systems have been and are being developed on the basis of D2S. Examples are the Dial Your Disc (DYD)-system, which presents information in English about Mozart compositions (Deemter et al., 1994); (Collier and Landsbergen, 1995), and the GoalGetter system, which presents spoken monologues in Dutch about the course and the result of a football game (Klabbers et al., 1997b). In this paper, we illustrate the prosodic rules used in D2S with examples from GoalGetter.

After a brief description and illustration of the general architecture of D2S, we describe in detail how the prosodic component of D2S computes the prosodic properties of the generated sentences. Then we discuss how the resulting prosodic annotations are used in the various speech output techniques employed in D2S. We end with some remarks about evaluation of the prosodic rules and a conclusion.

2 Architecture of D2S

The general architecture of D2S is represented in Figure 1. It consists of two modules, the *Language Generation Module (LGM)*, and the *Speech Generation Module* (SGM).

The LGM takes data as input and produces *enriched text*, i.e., prosodically annotated text. For instance, it contains annotations to indicate accents and prosodic boundaries. This is input to the SGM, which turns it into a speech signal.

Our example system GoalGetter (Klabbers et al., 1997b) takes data on a football match as input. The output of the system is a correctly pronounced, coherent monologue in Dutch which conveys the infor-

*Authors Theune and Klabbers carried out this research within the framework of the Priority Programme Language and Speech Technology (TST). The TST-programme is sponsored by NWO (Dutch Organization for Scientific Research).

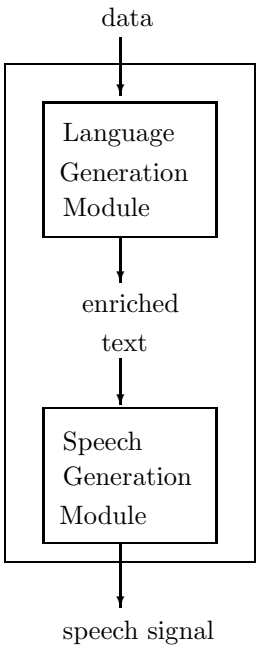


Figure 1: Global architecture of D2S

mation on this match. An example of the input data is given in Figure 2, and one possible output text is given in Figure 3. In the enriched text, pitch accents are indicated by double quotes (") and phrase boundaries of varying strength are indicated by one to three slashes (/). The other symbols used in the text will be clarified in Section 4.

team 1:	PSV
goals 1:	1
team 2:	Ajax
goals 2:	3
goal 2:	Kluivert (5)
goal 2:	Kluivert (18)
goal 2:	Blind (83/pen)
goal 1:	Nilis (90)
referee:	Van Dijk
spectators:	25.000
yellow 1:	Valckx

Figure 2: Example input of the LGM

Since we lack the space for a full description of the LGM, presented schematically in Figure 4, we only point out some important aspects which are relevant for the prosodic rules given in Section 3. For a more detailed description, see Klabbers et al. (1997a).

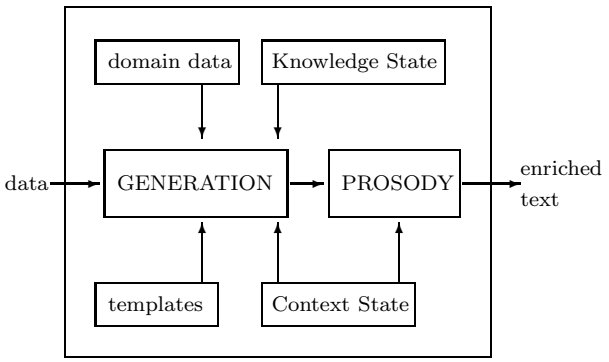


Figure 4: The architecture of the Language Generation Module (LGM)

The input for the LGM consists of *data* on specific football matches (see Figure 2) and on the domain, e.g., background information on the players and the teams. The information in the input can be expressed by means of *templates* in the form of a syntactic tree with variable slots. Choice and ordering of the templates and the filling of their slots depend on conditions on (1) the *Knowledge State*, which keeps track of which information has been expressed, and (2) the *Context State*, in which various aspects of the context are represented (Deemter and Odijk, 1995).

A central part of the Context State is the *Discourse Model*, which contains information about which linguistic expressions have been used in the preceding text. Rules formulated in terms of this Discourse Model make it possible to use various referential expressions (proper names, pronouns, definite descriptions, etc.) appropriately. For instance, in the fourth sentence of the example text given in Figure 3, *Dertien minuten later liet de aanvaller zijn tweede doelpunt aantekenen* ('Thirteen minutes later the forward had his second goal noted'), it was possible to use a definite description (*de aanvaller*, 'the forward') to refer to Kluivert, because the Discourse Model contained an appropriate unique antecedent (namely, the proper name *Kluivert* that was used in the third sentence). When a new sentence has been generated, the Discourse Model is updated accordingly, and then the sentence with its full parse tree and the updated Discourse Model are input to the *Prosody* module.

3 Computing prosody

In this section we present the rules that are used in the Prosody module of the LGM, which determines the location of accents and phrase boundaries in a generated sentence on the basis of both syntactic

"De "wedstrijd tussen "PSV en "Ajax / eindigde in "@een // - "@drie /// "Vijfentwintig duizend "toeschouwers / bezochten het "Philipsstadion /// "Ajax nam na "vijf "minuten de "leiding / door een "treffer van "Kluivert /// "Dertien minuten "later / liet de aanvaller zijn "tweede doelpunt aantekenen /// De % "verdediger "Blind / verzilverde in de "drieentachtigste minuut een "strafschop voor Ajax /// "Vlak voor het "eindsignaal / bepaalde "Nilis van "PSV de "eindstand / op "@een // - "@drie /// % "Scheidsrechter van "Dijk / "leidde het duel /// "Valckx van "PSV kreeg een "gele "kaart ///	Translation: The match between PSV and Ajax ended in 1-3. Twenty-five thousand spectators visited the Philips stadium. After five minutes, Ajax took the lead through a goal by Kluivert. Thirteen minutes later the forward had his second goal noted. The defender Blind kicked a penalty home for Ajax in the 83rd minute. Just before the end signal, Nilis of PSV brought the final score to 1-3. Referee Van Dijk led the match. Valckx of PSV received a yellow card.

Figure 3: Example output of the LGM

and semantic information. First we will discuss the accentuation algorithm, which is based on a version of Focus-Accent Theory proposed in Dirksen (1992) and Dirksen and Quené (1993). In Focus-Accent Theory, binary branching metrical trees are used to represent the relative prominence of nodes with respect to pitch accent.

We will use our previous example sentence, *Dertien minuten later liet de aanvaller zijn tweede doelpunt aantekenen*, as an illustration. First, the accentuation algorithm constructs the sentence's metrical tree, shown in Figure 5 (simplified). In our implementation, this tree corresponds to the sentence's syntactic tree,[1] except that its nodes have *focus* markers and are labeled *weak* or *strong*. The focus properties of the nodes in the metrical tree are determined as follows.

Initially, all maximal projections (NP, VP etc.) are assigned a positive focus value, indicated as [+F]. The other nodes are not specified for focus. These initial focus values can be changed by non-syntactic factors causing the focus value to become negative, indicated as [-F]. This happens in three cases: (1) a node dominates an unaccentable word; (2) a node represents given information;[2] (3) a node dominates only nodes which are marked [-F]. Unaccentable words, e.g., certain function words, are explicitly listed. Our example sentence contains only one such word, the determiner *de* ('the'). The rules for determining givenness are based on the theory proposed by van Deemter (1994), who distinguishes two kinds of givenness: object-givenness and concept-givenness.

A phrase is regarded as object-given if it refers to a discourse entity that has been referred to earlier in its local discourse domain, which in the present implementation consists of all preceding sentences in the same paragraph. In the example, checking the Discourse Model reveals that the phrases *de aanvaller* ('the forward') and *zijn* ('his') are object-given, because their referent (Kluivert) was referred to in the preceding sentence, which belongs to the same paragraph. This means that their dominating nodes in the metrical tree must be marked [-F]. This example illustrates that object-givenness does not depend on the surface form of the referring expression, but only on its referent. The expressions *de aanvaller* and *zijn* are object-given even though they were not used earlier in the text.

The second kind of givenness, concept-givenness, occurs if the root of a word is synonymous (including identity) with the root of a preceding word in the local discourse domain, or if the concept expressed by the second word subsumes the concept expressed by the first word. Our example sentence contains two instances of the first case: the words *minuten* and *doelpunt* are concept-given, and therefore marked [-F], due to the presence in the preceding sentence of

[1] Unary branching of metrical trees is allowed.

[2] This is based on the observation by Halliday (1967), Chafe (1976), Brown (1983) and others that phrases expressing 'new' information are normally accented, while phrases expressing 'given' or 'old' information are usually deaccented.

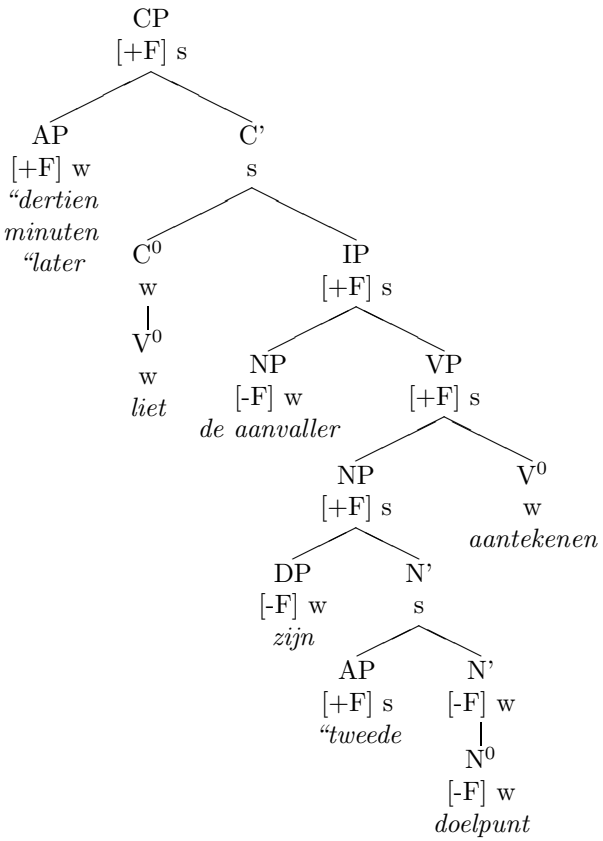


Figure 5: Metrical tree for the fourth sentence.

the synonymous words *minuten* and *treffer* respectively. The second case, subsumption, can be illustrated by the sequence *Kluivert is een heel goede aanvaller; Hij is de beste speler van Ajax* ('Kluivert is a very good forward; He is the best player of Ajax'). Since the concept **speler** ('player') subsumes the concept **aanvaller** ('forward'), the word *speler* in the second sentence will be defocused due to concept-givenness.

Note that the first case of concept-givenness is the only kind of givenness distinguished in D2S which can also be determined in a relatively easy way in unrestricted Text-to-Speech systems, e.g., NewSpeak (Hirschberg, 1990); (Hirschberg, 1992). The second case of concept-givenness, subsumption, will be very difficult to detect in an unrestricted Text-to-Speech system because it requires the presence of a concept hierarchy, which is only feasible if the relevant concepts are known in advance. Finally, determining object-givenness will also be very difficult in Text-to-Speech, because it makes very high demands on text analysis.

After the metrical tree nodes have been assigned focus markings, their weak/strong labelling can be determined. This labelling depends both on the structure of the tree and the focus properties of the nodes. In Dutch, the structural rule is that the left node of two sisters is weak and the right node is strong, unless the right node is a zero projection, like the V^0 node dominating *aantekenen* in figure 5.[3] This structural labelling can be changed under the influence of focus. If the structurally strong node is marked [-F] while the structurally weak node is not, the so-called Default Accent Rule applies and the labelling is switched. In figure 5, this happened to the AP dominating *tweede* and the N' dominating *doelpunt*. The N' is marked [-F] because all the nodes it dominates are marked [-F]. (See defocusing rule (3) given above.)

After the weak/strong labelling has been determined, accents are assigned according to the following algorithm: each node that is marked [+F] launches an accent, which trickles down the tree along a path of strong nodes until it lands on a terminal node (a word). In our example, the accents launched by CP, IP and VP all coincide with the accent launched by the NP node dominating *zijn tweede doelpunt*, finally landing on the word *tweede*. Note that if the word *doelpunt* had not been concept-given, then the N^0 and the N' would not have been marked [-F] and the Default Accent Rule would not have applied. The accent would then have landed on *doelpunt*.

Since the NP node dominating *de aanvaller* is weak, no accent trickles down to it, and because it is marked [-F] it does not launch an accent itself. The AP node dominating the phrase *dertien minuten later* (its internal structure is not shown due to lack of space) does launch an accent, which trickles down to the word *later*. The NP *dertien minuten*, which is contained in the AP, also launches an accent; since this cannot land on the word *minuten* (which is defocused due to concept-givenness) it ends up on the word *dertien*.

Recently, an algorithm for the generation of contrastive accent has been added to the GoalGetter system. This algorithm assigns a pitch accent to phrases which provide contrastive information, over-

[3]Evidence for this rule comes from constructions like the following:
(i) Kluivert liep [$_{VP}$ [$_{P^0}$ voorbij] [$_{NP}$ het doel]]
(ii) Kluivert liep [$_{VP}$ [$_{NP}$ het doel] [$_{P^0}$ voorbij]]
Both (i) and (ii) can be translated as 'Kluivert walked past the goal'. Since *voorbij* is not accented in either case, the P^0 node should be labeled weak. The fact that *voorbij* is unaccentable in these positions cannot be explained by claiming the word itself is unaccentable, since in *Kluivert liep er voorbij* ('Kluivert walked past it') the word does receive an accent.

riding deaccentuation due to givenness. For more details on the algorithm, see Theune (1997).

After accentuation, phrase boundaries are assigned. Three phrase boundary strengths are distinguished.[4] The sentence-final boundary (///) is the strongest one. Words which are clause final (i.e., the last word in a CP or IP) or which precede a punctuation symbol other than a comma (e.g., ';') are followed by a major boundary (//). Minor boundaries (/) are assigned to words preceding a comma. Additionally, constituents to the left of an I', a C' or a maximal projection are followed by a weak boundary, provided that both constituents are accessible for accent, and that the left one has sufficient length (more than four syllables). This is a slightly modified version of a structural condition proposed by Dirksen and Quené (1993). In our example only the AP *dertien minuten later* meets this condition and is therefore followed by a minor phrase boundary. Since the sentence contains no punctuation and consists of just one clause, the only other phrase boundary is the sentence-final one.

4 Speech Generation

The SGM has two output modes, phrase concatenation and phonetics-to-speech, each of which makes optimal use of the prosodic markers generated by the LGM. We start with a brief description of the two output modes, followed by a discussion of the prosodic realization in either output mode.

Phrase concatenation - Phrase concatenation is a technique which tries to reconcile the high-fidelity quality and inherent naturalness of prerecorded speech with the flexibility of synthetic speech. Entire phrases and words are recorded, and played back in different orders to form complete utterances. In this way a large number of utterances can be pronounced on the basis of a limited number of prerecorded phrases, saving memory space and increasing flexibility. This technique is best applied to a carrier- and slot situation where there is a limited number of types of utterances (carriers) with variable information to be inserted in fixed positions (slots). The systems based on D2S fit this situation well. The carriers correspond to the syntactic templates and these have slots for variable information such as match results, player names, etc.

Successful application of the phrase concatenation technique is not quite as trivial as it may seem at first sight. If all the phrases are recorded in isolation without taking their accentuation or their position in the sentence into account, the resulting speech will have discontinuities in duration, loudness and intonation. Our method is more sophisticated in that different prosodic variants for otherwise identical phrases have been recorded. To determine how many and what prosodic realizations should be recorded for each phrase, a thorough analysis of the material the system can generate is required.

Phonetics-to-Speech - Synthetic speech is far more flexible than any form of prerecorded speech. Since there is complete control over the realization it is very well suited to test the accentuation and phrasing rules. In commercial applications synthetic speech is not used very often since the naturalness of the output speech still leaves a great deal to be desired.

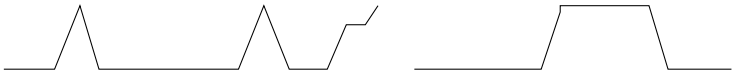
Because the LGM provides all relevant information there is no need for full-fledged text-to-speech synthesis. The LGM generates an orthographic representation which has a unique mapping to a phonetic representation.[5] This makes it possible to do errorless grapheme-to-phoneme conversion by looking up the words in a lexicon instead of using rules. Our phonetics-to-speech system, SPENGI (SPeech synthesis ENGIne) uses diphone concatenation in either LPC or PSOLA format. The rule formalism for intonation is an implementation based on the intonation theory of 't Hart et al. (1990).

Realizing prosody in speech generation - The enriched text that the LGM generates contains several prosodic markers. In the phrase concatenation component these markers trigger the choice of the appropriate prosodic variant from the phrase database and the pauses to be inserted at the appropriate positions.

The carrier sentences have been recorded in just one prosodic version. The variable words that fill the slots have been recorded in six different prosodic variants to account for the place in the sentence where they occur and the accentuation they receive. A word can be either accented or deaccented. We did not instruct our speaker as to how to realize the accents in the carrier sentences. In the variables we just made sure that accents were realized consistently in each category. When a word occurs before a minor phrase boundary the word is realized with a continuation rise. A major phrase boundary triggers a pause and possibly a lengthening of the word preceding the boundary. Before a final phrase boundary, the word is realized with a final fall. Inserting the right words in the right contexts optimizes the

[4] In longer texts, containing more complicated constructions, it might be desirable to distinguish more levels. Sanderman (1996) proposes a boundary depth of five to achieve more natural phrasing.

[5] It could also generate a phonetic representation directly.



de "wedstrijd tussen "psv en "ajax / "eindigde in "@een // "@drie ///

Figure 6: Stylized pitch contour of the introductory sentence

prosody of the output speech, thus achieving fluency and a natural rendering.

In Dutch, the score of a match is pronounced in a special way: the major boundary between the two numbers triggers lengthening of the first number and a pause between the two numbers, but the two accented numbers are realized with a so-called 'flat hat' pattern as if they were part of the same clause (see 't Hart (1990) for a description of pitch movements). This is indicated by a special marker used only in the phrase concatenation component of GoalGetter (the @-sign). There is another special marker (the %-sign) to mark nouns functioning as an adjunct to another noun. The special nouns are always accented and shorter in duration than when they occur as a head noun. Figure 6 shows a stylized pitch contour of the opening sentence of Figure 3, which illustrates how the score is pronounced.

In the phonetics-to-speech component the prosodic markers are used to trigger the intonation and duration rules. Intonation is represented as a series of pitch movements with restrictions on the possible combinations of movements. The words that are accented are given a prominence-lending pitch pattern (a pointed hat or a flat hat are most commonly used). At the boundaries a pause of some length can occur, where the length of the pause depends on the strength of the boundary. A boundary can also trigger a continuation rise or pre-boundary lengthening, as mentioned above. To allow for variation in the intonation, each rule has a number of weighted alternatives from which a random choice is made (taking the weights into account). This also makes it possible to have some optional rules, for instance, for the melodic highlighting of syntactic boundaries which is not obligatory.

5 Evaluation

Nachtegaal (1997) reports on a small experiment which was carried out to test the accentuation algorithm of D2S. No formal evaluation has taken place for the algorithm determining the placement of phrase boundaries.

In the experiment by Nachtegaal (1997), Dutch speakers were asked to read aloud texts generated by the LGM of GoalGetter. Recordings of the read texts were presented to 'expert listeners' who indicated on which words they heard an accent. Comparisons were then made between the accentuation patterns produced by the speakers and those generated by the system. The results of the experiment were positive: the number of words on which the accentuation by the speakers deviated from the accentuation by GoalGetter was very small (less than 4% of all accentable words, i.e., excluding 'unaccentables' like function words etc.). The texts used in the experiment contained sentences which were structurally similar to those of the example text given in Figure 3. Not all syntactic constructions which are currently generated by GoalGetter were included in the test. The prosody of the current version of GoalGetter was only evaluated informally, but the results were in line with those of Nachtegaal (1997).

The prosodic rules described in this paper have also been succesfully implemented in the DYD-system (Deemter et al., 1994); (Collier and Landsbergen, 1995), which differs from GoalGetter with respect to language (English versus Dutch) and domain (Mozart compositions versus football reports). Informal evaluation of the prosody in DYD gave equally satisfactory results as for GoalGetter. This was as expected, since the prosodic rules of D2S are essentially domain- and language independent.[6] All things considered, the quality of the prosodic rules of D2S is judged to be good.

6 Conclusion

To determine the prosodic properties of a sentence in a text, information about both sentence-internal and sentence-external syntax and semantics is needed. In Text-to-Speech this information has to be obtained through text analysis, whereas in Data-to-Speech reliable information of this kind is readily available. As a consequence, Data-to-Speech provides a better basis for using sophisticated prosody assignment rules than Text-to-Speech.

[6] Only the rule governing the weak/strong labelling of the metrical tree nodes has a language-specific parameter.

The prosodic rules discussed in this paper are implemented in a generic Data-to-Speech system called D2S. They make crucial use of both sentence-internal and sentence-external semantic and syntactic information, provided by the system in the form of a Discourse Model and parse trees of the generated sentences. The reliable and detailed character of this information makes it possible to assign prosodic markings which are reliable and detailed as well. This in turn allows for a close control of prosodic realization, resulting in natural-sounding intonation.

References

Brown, G. 1983. Prosodic structure and the given/new distinction. In D. R. Ladd and A. Cutler, editors, *Prosody: Models and Measurements*. Springer Verlag, Berlin.

Chafe, W.L. 1976. Givenness, contrastiveness, definiteness, subjects, topics and points of view. In C. N. Li, editor, *Subject and Topic*. Academic Press, New York.

Coch, J. 1996. Evaluating and comparing three text-production techniques. In *Proceedings COLING 1996*, pages 249–254.

Collier, R. and J. Landsbergen. 1995. Language and speech generation. *Philips Journal of Research*, 49(4):419–437.

Deemter, K. van. 1994. What's new? A semantic perspective on sentence accent. *Journal of Semantics*, 11:1–31. CSLI report No. CSLI-93-178.

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

Deemter, K. van and J. Odijk. 1995. Context modeling and the generation of spoken discourse. Manuscript 1125, IPO, Eindhoven. Philips Research Manuscript NL-MS 18 728, to appear in *Speech Communication* 21 (1/2).

Dirksen, A. 1992. Accenting and deaccenting: A declarative approach. In *Proceedings of COLING 1992, Nantes, France*. IPO MS. 867.

Dirksen, A. and H. Quené. 1993. Prosodic analysis: The next generation. In van Heuven and Pols, editors, *Analysis and Synthesis of Speech: Strategic Research Towards High-Quality Text-to-Speech Generation*. Mouton de Gruyter, Berlin - New York.

Halliday, M.A.K. 1967. Notes on transitivity and theme in English. *Journal of linguistics*, 3:199–244.

Hirschberg, J. 1990. Accent and discourse context: assigning pitch accent in synthetic speech. In *Proceedings of the 8th National Conference on Artificial Intelligence, Menlo Park, 29 July - 3 August, 1990*, pages 952–957. MIT Press.

Hirschberg, J. 1992. Using discourse context to guide pitch accent decisions in synthetic speech. In G. Bailly, C. Benoît, and T.R. Sawallis, editors, *Talking Machines: Theories, Models and Designs*. Elsevier Science Publishers B.V., pages 367–376.

Klabbers, E., J. Odijk, J.R. de Pijper, and M. Theune. 1997a. From data to speech: A generic approach. IPO MS 1202.

Klabbers, E., J. Odijk, J.R. de Pijper, and M. Theune. 1997b. GoalGetter: From Teletext to speech. To appear in IPO Annual Progress Report 31, 1996.

Nachtegaal, D. 1997. An evaluation of GoalGetter's accentuation. Report 1142, IPO, Eindhoven.

Reiter, E. 1995. NLG vs. templates. In *Proceedings of the Fifth European Workshop on Natural Language Generation*, pages 95–106, Leiden, 20-22 May. University of Leiden.

Sanderman, A. 1996. *Prosodic Phrasing: production, perception, acceptability and comprehension*. Ph.D. thesis, Eindhoven University, Eindhoven.

't Hart, J., R. Collier, and A. Cohen. 1990. *A Perceptual Study of Intonation: an Experimental Phonetic Approach to Speech Melody*. Cambridge University Press, Cambridge.

Theune, M. 1997. Contrastive accent in a data-to-speech system. In *Proceedings ACL/EACL 1997*. To appear.