Improving System Output Using the Information State

Ivana Kruijff-Korbayová Stina Ericsson
Carlos García Rebecca Jonson Elena Karagjosova
Pilar Manchón Kepa J. Rodríguez José F. Quesada

Distribution: PUBLIC

Specification, Interaction and Reconfiguration in Dialogue Understanding Systems
IST-1999-10516

Deliverable D5.1

October 2002

IST-1999-10516 SIRIDUS

Specification, Interaction and Reconfiguration in Dialogue Understanding Systems

Göteborg University
Department of Linguistics

Linguamatics Ltd

Telefónica Investigación y Desarrollo SA Unipersonal
Speech Technology Division

Universität des Saarlandes
Department of Computational Linguistics

Universidad de Sevilla
Departamento de Lengua Inglesa

For copies of reports, updates on project activities and other SIRIDUS-related information, please look on our website at www.ling.gu.se/projekt/siridus.

For technical matters, please contact the Technical Coordinator, Robin Cooper and for administrative matters, please contact the Administrative Coordinator, Mareike Schmitt.

Prof. Robin Cooper
Department of Linguistics
Gothenburg University
Box 200
SE-405 30 Gothenburg
Sweden

Phone: +46 31 773 2536
Fax: +46 31 773 4853
cooper@ling.gu.se

Mareike Schmitt
European Project Office
Saarland University
c/o EURICE GmbH
Science Park Saar
Stuhlsatzenhausweg 69
D-66123 Saarbrucken
Germany
Phone: +49 (0) 681 959 233 66
Fax: +49 (0) 681 959 233 70
ms@eurice.de

Primary responsibility for authorship is divided as follows:

- Chapter 1: I. Kruijff-Korbayová
- Chapter 2: I. Kruijff-Korbayová and P. Manchón (Section 2.2)
- Chapter 3: I. Kruijff-Korbayová
- Chapter 4: I. Kruijff-Korbayová and S. Ericsson (Section 4.3)
- Chapter 5: S. Ericsson
- Chapter 6: I. Kruijff-Korbayová, E. Karagjosova and K. J. Rodrigues
- Chapter 7: P. Manchón and J. F. Quesada
- Chapter 8: P. Manchón (Section 8.1), I. Kruijff-Korbayová (Section 8.2), R. Jonson and C. García (Section 8.3)
- Chapter 9: E. Karagjosova and K. J. Rodrigues (Section 9.1), R. Jonson and C. García (Section 9.2), P. Manchón (Section 9.3)
- Chapter 10: I. Kruijff-Korbayová, S. Ericsson, C. García, P. Manchón

I. Kruijff-Korbayová was the editor.

Contents

Chapter 1

Introduction

Our goal in this report is to explore the use of the dialogue context model as represented in the information state of a dialogue system to control variations in the realization of the system's output. Our main concern is contextually appropriate variation of prosodic realization of the spoken output, but we also discuss other aspects of surface realization, such as contextually appropriate choice of syntactic structure and word order, and the choice between full vs. short utterances.

Generation of natural synthetic speech is one of the challenges of spoken dialogue systems aiming at human-like interaction face. In SIRIDUS, we are concerned with systems which allow relatively high degree of flexibility in the way communication proceeds. Therefore, "sentences", i.e., strings of words, with the same propositional content are uttered by the system in various contexts. Different contexts, however, require different intonation to be used in order for an utterance to sound natural and contextually appropriate. The use of the same (default) intonation in different contexts may have negative effect of intelligibility of the system's output, for example when the intonation of an answer does not correspond to the question asked. It is suspected that inappropriate intonation of system utterances may in extreme cases even lead to user's confusion.

However, the details of relating prosody and other aspects of realization to various aspects of context are still very much a research topic. While we cannot hope to resolve all the open questions and present a complete account, our aim in the present work is to make a contribution anchored in an implemented end-to-end system, which enables concrete experiments.

In Chapter 2, we present a range of examples that substantiate the claim that different

contexts require different intonation to be used in order for an utterance to sound natural and contextually appropriate. More generally, the prosodic realization as a whole needs to be controlled taking dialogue context into account. We discuss several factors involved in producing contextually appropriate synthetic speech, including information structure, dialogue progress history, dialogue move expectations, intelligent barge-in and manipulations of word pronunciation in context.

Of these factors, information structure is the one we address in detail in this report. Information structure is an inherent aspect of utterance meaning, which reflects a partitioning of meaning according to how an utterance relates to the context and how it updates it. At the same time, information structure is a level of meaning which unifies a range of interacting contextually-dependent aspects of utterance realization, including intonation, word order, syntactic constructions, morphological marking, and choice of shortened forms of expressions, such as anaphora and ellipsis. These realization means are encountered in various combinations within different languages as well as cross-linguistically. Determining the information structure partitioning of system utterances according to the context, and producing the corresponding realizations, are therefore important steps towards generating natural and contextually appropriate system output.

In Chapter 3, we briefly overview the varied landscape of approaches to information structure, and we motivate and explain the basic notions we use in our current work. The amount to information structure partitioning in two dimensions, namely a Theme/Rheme partitioning which reflects an aboutness relation, and a Background/Focus partitioning which reflects contrast between contextually relevant alternatives.

Although our discussion in this report mostly concerns issues of contextual appropriateness of intonation, intonation is not the only aspect in the realization of system utterance which is sensitive to context, and it is also only one of the means of realizing information structure. We discuss intonation, word order and shortened utterances as possible reflexes of information structure in Chapter 4.

In Chapter 5 we address the relation between information structure and context in detail. Building on earlier work in the TRINDI project (Engdahl et al., 2000), we define a set of rules which specify how an information structure partitioning of utterance meaning can be derived from the information state of dialogue participants, which is used to represent context in the information-state update approach to dialogue developed in the SIRIDUS project. Our rules capture the following ideas: the Theme/Rheme partitioning is derived on the basis of the current question under discussion; the Background/Focus partitioning is obtained by comparing the current propositional content with relevant similes, found either in the shared commitments part of the information state or in the representation of the domain.

On the basis of these specifications, an experimental implementation in the GoDiS system

has been developed, which we present in Chapter 6. The rules assigning information structure are implemented as a module that takes as input the propositional content of a dialogue move, and returns this content partitioned according to the current question under discussion, and the contents of the shared commitments and the domain knowledge. The partitioned content that this module outputs serves as input to the generation of the surface realization, which produces a string of words along with an annotation of the information structure partitioning. This annotation uses an internal set of labels. These are subsequently converted into markup suitable for text to speech synthesis.

Before we discuss the details of producing contextually varied spoken output using various speech synthesis systems, we first give a brief background about the state of art in speech synthesis in Chapter 7. Here we overview the basic strategies and techniques used in speech synthesis and the speech markup standards. The use of markup standards highly facilitates the integration of off-the-shelf synthesis systems in a modular way. However, the available standards such as SABLE or SSML do not (yet) support higher level intonation annotation, such as ToBI. There are two exceptions to this among the publicly available text-to-speech synthesis systems we are aware of, namely the FESTIVAL system for English which supports ToBI in an experimental version, and the MARY system for German which supports the German ToBI. Both systems also support the SABLE standard. However, it is the possibility of using ToBI that makes them particularly suitable for experimenting with contextually varied intonation.

In Chapter 8 we provide a short introduction to the text-to-speech synthesis systems used in our implementations, namely FESTIVAL, MARY and Telefónica's TTS.

In Chapter 9 we describe how contextually varied speech output is produced using off-the-shelf speech synthesis systems in various versions of the dialogue systems developed in the SIRIDUS project. First, we present varied intonation production in GoDiS based on the information structure partitioning assignment described in Chapters 5 and 6. We integrated both FESTIVAL and MARY in GoDiS, and defined various mappings from our internal information structure annotation to intonation annotation formats used by these systems. This enables us to experiment with the following different versions: MARY for German with either SABLE or GToBI intonation annotation, MARY for English with SABLE intonation annotation, and FESTIVAL for English with either SABLE or ToBI annotation. We present examples of input/output annotations for these different versions. The corresponding wave files can be accessed through *http://www.coli.uni-sb.de/cl/projects/Siridus/*.

We also discuss the production of varied speech output in Spanish using Telefónica's TTS, and the design of a set of strategies for experimenting with and evaluation of varied speech output in the Delfos 2 system using FESTIVAL.

Finally, Chapter 10 concludes the report with a summary of the achieved results and a list of suggestions for future work.

Chapter 2

Factors in Generation of Natural Synthetic Speech

Generation of natural synthetic speech is one of the challenges of spoken dialogue systems aiming at human-like interaction face. Synthetic speech must not only be intelligible, but also natural. But how is naturalness defined? Naturalness of spoken output cannot be considered only from a *static perspective*, i.e., the generation of naturally sounding words or sentences in isolation, but also from a *dynamic perspective*, i.e., the generation of synthetic speech appropriate in the given dialogue context.

To introduce the problem, let us start from the common linguistic assumption that for each string of words ("sentence") there exists an unmarked, neutral intonation pattern (or more generally: prosodic realization), i.e., the intonation pattern that a human would use "out of the blue" (without assuming any particular context shared between her and the hearer). For example, consider the sentences below:

(1) When do you want to leave?
(2) When do you want to leave Prague?
(3) The price is five hundred Euro.
(4) You can leave from Prague.

When a human utters them "out of the blue", she places the nuclear intonation center as indicated by SMALLCAPS below:

(1′) When do you want to LEAVE?

(2′) When do you want to leave PRAGUE?

(3′) The price is FIVE HUNDRED EURO.

(4′) You can leave from PRAGUE.

What does an off-the-shelf text-to-speech (TTS) synthesis system produce? Without any additional information about the required prosodic realization, a TTS synthesis system uses some default. In an ideal case, this default might be expected to correspond to the unmarked/neutral intonation indicated above. However, this expectation is not born out in reality.

We have observed this with several demonstration versions of TTS systems for English available on-line. For example, for the sentence *When do you want to leave?*, none of the systems we tried produced (1′) The "default" outputs they did produce are sketched below.

(5) a. When do you WANT to leave? — Festival[1] and ViaVoice[2]

b. When DO you want to leave? — Lucent's Articulator[3]

c. When do you want TO leave? — AT&T's TTS[4]

Not only are these not unmarked/neutral. The output in (5a) is unnatural or even misleading (because it implicates a contrast between wanting and some other attitude), the output in (5b) is misleading (because it implicates a contrast between times when the user wants and does not want to travel), and the output in (5c) is outright wrong.

Therefore, when using off-the-shelf TTS synthesizers, it cannot be assumed that the default intonation a synthesizer produces actually corresponds to the unmarked/neutral intonation. What this means is that even in seemingly simple cases we cannot rely on the synthesizer's defaults Moreover, (5) also shows that different synthesizers have different defaults. Therefore, if we want to be able to switch between different synthesizers without having to make changes in the rest of the system, we can't assume any default.

As a counter-argument to the above conclusion, one could point out that most dialogue systems are domain dependent. It is this domain dependency among other factors that allows for the use of restricted vocabulary, knowledge and language models. For the time being, dialogue systems are even designed to deal with a finite set of tasks and situations. Within their particular domain, there is usually a finite set of structures, responses and situations that the system is ready to handle, and this set usually has dominant types of

[1] `festvox.org/voicedemos.html` demo version October 2002.

[2] `www-3.ibm.com/software/speech/enterprise/dcenter/demo-tts.html`; demo version March 2002.

[3] `www.tts-talk.com`; demo version March 2002.

[4] `www.naturalvoices.att.com/demos`; ; demo version March 2002.

utterances (intonation) that are more contextually appropriate. Taking into account the particular domain where the dialogue is taking place is therefore relevant to determine what prosodic patterns are more adequate. Therefore, one could argue that instead of using an off-the-shelf TTS synthesis system, one could use a system that has been trained to produce defaults suitable for the particular domain of application.

This might be a solution for a system with a very restricted range of produced output structures, and a completely fixed range of contexts in which these outputs are produced. However, what we are concerned with are systems which allow flexibility, and therefore the same "sentences" can be uttered by the system in various contexts. And as we shall see below, the appropriateness of any particular intonation varies in different contexts, in other words, different contexts may require different prosodic realizations in order for an utterance to sound natural and contextually appropriate.

Since it cannot be assumed that any given intonation will be sufficient/satisfactory in all contexts, we will therefore conclude that it is generally necessary to control the placement of the intonation center(s) in every system utterance. More generally, the prosodic realization as a whole needs to be controlled, as we shall see in more detail later.

We discuss some of the factors that influence contextual appropriateness of particular intonation patterns in more details in the next sections. First, we illustrate semantic factors such as question-answer congruence and contrast. Then we illustrate other factors, including dialogue progress history, dialogue move expectations, intelligent barge-in and manipulations of word pronunciation in context.

2.1 Semantic Factors

Differences in intonation, in particular, differences in the placement of pitch accents, can reflect differences in utterance meaning. This is one of the reasons why intonation needs to be varied in different contexts. We now support this claim by several examples. We first present examples that illustrate that the placement of the nuclear intonation center in an utterance that answers a question needs to be congruent with the question. Second, we present examples that illustrate the use of additional intonation centers to indicate contrast.

2.1.1 Question-Answer Congruence

We can observe that in coherent direct answers to questions, the nuclear intonation center coincides with that part of the answer-utterance that provides the information requested in the question. First, consider the following set of questions a user may utter in a dialogue with a travel agent:

(6) a. U: When does Lufthansa fly to Malmö?
b. U: Which airline flies to Malmö on Saturday?
c. U: To which city does Lufthansa fly on Saturday?

All the questions in (6) can be answered by the "sentence" *Lufthansa flies to Malmö on Saturday*, i.e., the answers have the same propositional content. However, in each case, the answer needs to have a different intonation pattern, in particular, differently placed nuclear intonation center, in order to be appropriate as an answer to the question. Thus, (6a) is appropriately answered by (7a), (6b) by (7b) and (6c) by (7c). Any other combination is unnatural, incoherent.

(7) a. S: Lufthansa flies to Malmö on SATURDAY.
b. S: LUFTHANSA flies to Malmö on Saturday.
c. S: Lufthansa flies to MALMÖ on Saturday.

The contrast between the following two examples illustrates the same phenomenon once more, with more complex utterances:

(8) U: How many flights from Madrid to Brussels are there today and at what time?
S: There is ONE flight from Madrid to Brussels today at THREE P.M.

(9) U: Is there any flight from Madrid or Barcelona to Brussels that departs before four p.m. today?
S: There is one flight from MADRID to Brussels today at THREE P.M.

In (8), the primary information requested is the number of flights available and their departure time. In (9), however, the primary information requested not how many, but whether there is one viable possibility that matches the user's needs. Although the propositional content of the answers is the same, the intonation should be different in each case.

What the above examples illustrate is that a direct answer needs to be congruent with the respective question by placing the nuclear intonation center at the parts which correspond to the information requested in the question.

2.1.2 Contrast

Next consider another pair of examples, this time from the home-device domain.

(10) U: What is the status of the heaters?
S: The heater in the KITCHEN is ON.
The heater in the HALL is OFF.
The heater in the BATHROOM is OFF.

(11) U: What is the status of the devices in the kitchen?
S: The HEATER in the kitchen is ON.
The STOVE in the kitchen is OFF.
The TELEVISION in the kitchen is OFF.

The primary information requested in both (10) and (11) is the same, namely the status of some devices. That is what motivates placing the nuclear intonation center on *on/off* in every answer. But we can observe that it is very natural to place an additional pitch accent on some other part of each answer: on the locations of heaters in (10), but on the devices in (11). These additional accents do not seem to be necessary: the utterances are congruent answers to the given questions with or without them. However, in their absence, the answers are monotonous and unnatural.

The reason why the additional accents improve naturalness of the answers is that there is an inherent contrast present between the heaters in different locations in (10), and between different devices in the kitchen in (11). The additional pitch accents reflect this contrast.

What both the examples of question-answer congruence and of contrast illustrate is that in order to determine the intonation of a system utterance, we need to take into account the dialogue context. What concrete aspects of the dialogue context are to be taken into account and how, will be discussed in later chapters of this deliverable. Furthermore, we shall see that the variation we illustrated above as a variation of the placement of accents is only part of the story. There are other aspects of prosodic realization that need to be controlled, such as the type of accent and the placement and type of intonation boundaries.

2.1.3 Short Utterances

Although our discussion is concentrating on issues of contextual appropriateness of intonation, intonation is obviously not the only aspect in the realization of system responses which is sensitive to context. Reconsidering the examples above, it would be equally possible for the system to provide short answers instead of the full ones above.

(12) a. U: When does Lufthansa fly to Malmö?
S: On SATURDAY.
b. U: Which airline flies to Malmö on Saturday?
S: LUFTHANSA.
c. U: To which city does Lufthansa fly on Saturday?
S: MALMÖ.

Note that also in the case of short answers, there needs to be congruence: the short answer corresponds to the information requested in the question. It is worthwhile noting that shortening an answer to only the requested information is not always possible. For example, the answers in (10) cannot really be shortened except by using some kind of anaphoric expressions, such as *the kitchen one* instead of *the heater in the kitchen*. The answers in (11) can be shortened by leaving out the locations, as in (13).

(13) U: What is the status of the devices in the kitchen?
S: The HEATER is ON.
The STOVE is OFF.
The TELEVISION is OFF.

Finally, we should note that where in English it is predominantly intonation that varies depending on context, in other languages it may be something else that varies, e.g., word order in languages with higher degree of word order freedom than English.

In order to account for the interplay of these different factors within any given language, as well as in a cross-linguistic perspective, a level of meaning representation seems to be needed where these different aspects can be unified. We follow the line of work in which *Information Structure* is taken to be such a level of meaning representation. Information structure is seen as an abstract partitioning of utterance meaning, which is realized through an interplay of realization means, including intonation, word order, syntactic constructions, morphological marking, and can be seen as one factor motivating choice of shortened forms of expressions, such as anaphora and ellipsis. Languages vary in the ways they use and combine these aspects of realization in order to form contextually appropriate utterances.

However, the realization of information structure is not the only factor that makes use of these linguistic resources, and in particular of intonation. We briefly address some other contextual factors that affect intonation in the next section.

2.2 Other Factors

2.2.1 Dialogue Progress History

Taking into account the relation of the propositional content of an utterance to the dialogue history is very important in a dialogue system, as we have already seen in the previous section. In addition, dialogue history is necessary for anaphora resolution, to handle more than one task at the time, to keep track of what has just been done versus what needs to be done, to save processing time in certain situations, to be able to backtrack in case of error and retake the task without much ado, to learn from the corpus of conversations, to implement plan recognition strategies, etc.

What we want to point out here is that a dialogue system could take advantage of dialogue history in terms of whether preceding communication attempts were successful or not, in order to determine whether prosodic patterns other than the default designated for the particular task at hand are necessary. For instance, note the following dialogue:

(14) S: What's your user name? *(Default intonation)* (1)
U: Puppet Master
S: [Recognized: "Purchase Master"; Confidence level: Low]
Could you repeat your user name please? (2)
U Pu–ppet–Mas–ter [5]
S: [Recognized: "poo pit mass tear"; Confidence level: Low]
Hmm, I didn.t understand. Could you tell me your user name one more time? (3)

In (1), the intonation with which the question should be synthesized could be pretty much standard. Even in (2), the intonation could also be the default intonation. However, after having failed twice to recognize an utterance and even though the second error was partly caused by the user, from the human factors perspective it would be more polite to adopt

[5]Some users think that slowing down and breaking words into syllables will help the system recognize the utterance. Although this is a common human reaction, it actually makes the recognition impossible.

a more concessive or apologetic tone[6]. Therefore the intonation should not be that of a default question.

Although somewhat elaborated, the example above illustrates the case in which dialogue history would be essential to determine the prosodic pattern to be used in a rather sophisticated and human–factor–aware system. It also helps illustrate cases in which contextual help that also takes into account frequent HCI issues would be extremely useful.

2.2.2 Expectations

Given an architecture that allows the system to generate a set of expectations throughout the dialogue and rank them in terms of their likelihood, it would also be useful to make use of these expectations to convey the degree of certainty of the system about the dialogue.

In human–human dialogue, there are certain cues and prosodic patterns that convey more than simply what is being said. For instance, when given a certain context (dialogue history), one of the speakers is surprised by the turn of the conversation (broken expectations), the next utterance is usually predictable and either of the following:

- a confirmation of the communication line still being open,
- the utterance intonation, given the dialogue move triggered by the last move of the other speaker, conveys a certain degree of uncertainty that prompts the other speaker to confirm that the channel is still open.

Although these are very fine–grain observations and there are more significant elements to be implemented, we should nonetheless bear them in mind.

2.2.3 Intelligent Barge–In

Most dialogue systems do not handle feedback from the user in any form, and most (if not all) existing systems which handle barge-in will stop talking if they perceive any sound from the speaker. This means that even positive feedback (e.g. "uhuh") from the user will cause the system to stop speaking. This problem is aggravated in noisy environments, where noises or other people talking may be misinterpreted as speech from the user and cause a system to stop speaking.

[6]This is assuming that we are using a polite-friendly default personality for the system.

Thus instead of barge-in as interruption, what is needed is a concept of "Intelligent Barge-In", where the system not only allows the user to interrupt it, but attempts to recognize the new utterance to determines whether it is a genuine barge-in before interrupting the communication.

Mechanisms for handling overlapping user feedback have been explored within the GoDiS framework (Berman, 2001). This work aimed at the system making a distinction between different kinds of feedback from the user; positive feedback should usually not cause the system to stop speaking.

However, sometimes this is especially hard since the interruption may come with a bit of delay due to the time needed for the recognition process, making the dialogue a bit awkward. Other times, the system may not be entirely sure of whether it is or is not a real barge-in (medium-low confidence level). It is in these circumstances that the confirmation to continue or interrupt the dialogue should also have a more natural intonation that does not correspond to that of a normal question.

1. **SYSTEM[1]**: As an attention to our customers, we will send you the present of your choice for buying our latest product. Please select one of the following options: a DVD player, a HIFI stereo, (...)
2. **USER[1]**: *(addressing others while the system is enumerating the options)* I get to choose a present!
3. **SYSTEM[2]**: I didn't understand your answer. Please select one of the presents available: a DVD player, (...)
4. **OTHER[1]**: *(in the background)* Get the CD player!
5. **SYSTEM[3]**: *("DVD player", Confidence level: Medium)* Do you want to select the DVD player?
6. **USER[2]**: No. Continue with the list.
7. **SYSTEM[4]**: I'll continue with the list. Let me know which present you prefer when you are ready. You can also select a VCR, (...)
8. **OTHER[2]**: *(in the background)* Are you going to select the TV?
9. **SYSTEM[5]**: *(ignores lower confidence input)* (...) a frying machine, (...)

In **SYSTEM[3]**, the system perceives a plausible answer although the confidence level is only medium. Therefore the system, "knowing" that it has not finished reading the list, "knowing" that there have been previous unsuccessful interruptions and given the medium

confidence level, could generate a more appropriate prosodic contour for the confirmation. This special intonation is often used by human speakers when they are unsure of the answer, especially on the phone. This special intonation conveys to the user that the system "suspects" this might not be the right answer, increasing the user's confidence in the system and alleviating a potentially uncomfortable situation where the system is being interrupted over and over.

There are also certain contexts in which users are more likely than others to barge-in — for example when selecting one option out of a list, in which case the barge-in instant could also be used to estimate the most probable response. For these cases, the system could incorporate a set of heuristics that will help it decide on the course of action for the particular case.

Another interesting issue to be considered here is the use of back-channels in human dialogues. It would be necessary to be able to understand these utterances as what they are, rather than interrupting the dialogue each time they occur. On the other hand, we must also take into account that back-channels are not quite the same in HCI; humans tend to behave differently when they are aware of the fact that their counter speaker is a computer.

2.2.4 Pronunciation of words in context

Another aspect of prosodic realization is the pronunciation of individual words. Some experiments in this area seem to indicate that certain manipulations of the default synthesized forms taking context into account may render more natural utterance (Portillo, 1999), (Brinckmann and Trouvain, to appear).

Although most of these experiments were carried out with isolated words, tendencies to accept utterances with certain manipulations as more natural have been found. These manipulations imply the reduction of phoneme length and amplitude and the introduction of white noise. All manipulations were based on the notions of recognition point and phonemic entropy.

The figures below show how some of the manipulations were carried out.

It would be interesting to apply these concepts to phrases and intonation. It is a well-known fact that words may have different pronunciations depending on the immediate context, as well as their semantic role in the sentence and other factors such as utterance length and word-stress distribution. Since the Dialogue System may know in advance about all these factors, it would be possible to generate synthetic speech that would be prosodically appropriate for the circumstances, as well as more natural sounding.

Figure 2.1: Duration reduction

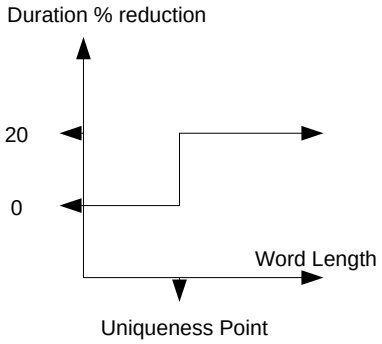
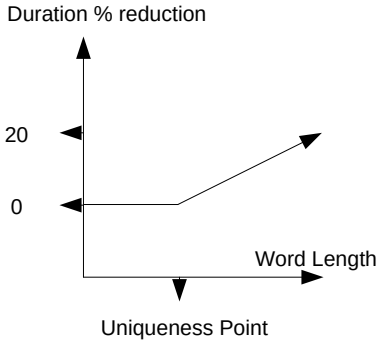


Figure 2.2: Progressive duration reduction



In order to do this, the input for the speech synthesizer must contain tags or labels that the synthesizer must be able to interpret. In other words, the system must incorporate:

1. Knowledge about all the factors above mentioned
2. Knowledge about how to interpret and use the relevant information from all modules to generate the appropriate intonation.
3. A mark-up language with enough expressive power to handle these issues
4. A synthesizer that can understand and act on those labels to produce the appropriate output.

2.2.5 Summary

In this chapter, we discussed several factors that have an impact on the appropriateness of the intonation of system utterances in various dialogue contexts. Of these, the semantic

factors which have to do with information structure, and were illustrated here by examples of question-answer congruence and of contrast, are central to the work described in this report. Information structure and it's use to improve the output of a dialogue system will be addressed in more detail in the following chapters of the present report: in Chapter 3 we present the basic notions of information structure in more detail, in Chapter 4 we discuss the realization of information structure by various means, in Chapter 5 we define rules that determine information-structure partitioning on the basis of the information state in GoDiS, in Chapter 6 we present an implementation of these definitions in GoDiS, and in Chapter 9 we describe how varied speech output is obtained in GoDiS, using off-the-shelf text-to-speech synthesis systems for English and German, and controlling the intonation of the output on the basis of the information structure partitioning.

Chapter 3

Information Structure

We present a brief introduction of the basic notions of *Information Structure* (IS). Because of the terminological variation among different approaches, but their close conceptual similarities, we do not discuss and/or compare individual approaches in detail (for comparative overviews, see for example (Kruijff, 2001), (Kruijff-Korbayová, 1998), (Vallduví, 1992)). Rather, we select one prominent approach, namely that proposed by Steedman (Steedman, 1996; Steedman, 2000a; Steedman, 2000b), we spell out how it relates to other prominent approaches and roughly align some of the different terminologies. Even though the discussion in the rest of this deliverable will be presented in terms of Steedman's account, we hope that the interested reader will be able to convert our proposal to the other theories if needed.

3.1 Basic Notions

Utterances in discourse or dialogue both *reflect* and *affect* the context: speakers organize their utterances in a way that reflects their model of the context (what they believe is shared between them and the hearer(s)) and the intended context change (corresponding to their communicative intentions). Thus, we can see some part(s) of an utterance meaning as being *context-dependent*, relating the utterance to the purpose of the discourse and anchoring it in the context (i.e., what speaker and hearer are attending to); other part(s) of an utterance meaning can be seen as *context-affecting*, advancing the discourse (i.e. adding or modifying some information). It is this division of utterance meaning into context-dependent and context-affecting parts that the notion of Information Structure tries to capture.

The terminology that is used in the literature to describe Information Structure and its

semantics is simultaneously various, and under-formalized. Yet it seems that all definitions have some elements in common. They all draw at least one of the following distinctions: (i) a "topic/comment" or "theme/rheme" distinction between the part of the utterance that relates it to the purpose of the discourse, and the part that advances the discourse; (ii) a "given/new" distinction, between parts of the utterance—actually, words—which contribute to distinguishing the content from other alternatives that the context makes available and those parts that are common to all of them.

There are differences among the theories of course. Some view these two distinctions as orthogonal, applying at independent levels of structure (Halliday, 1970a). Others in the Bolingerian tradition, view them as different aspects of a single level of structure.

There are further similarities: while some of the theories leave the discourse semantics associated with Information Structure at an intuitive level, the theories which do address formal semantic issues all tend to use some version of "update" semantics of either the Kampo-Heimian approach or the Alternative semantics approach following Rooth and Büring.

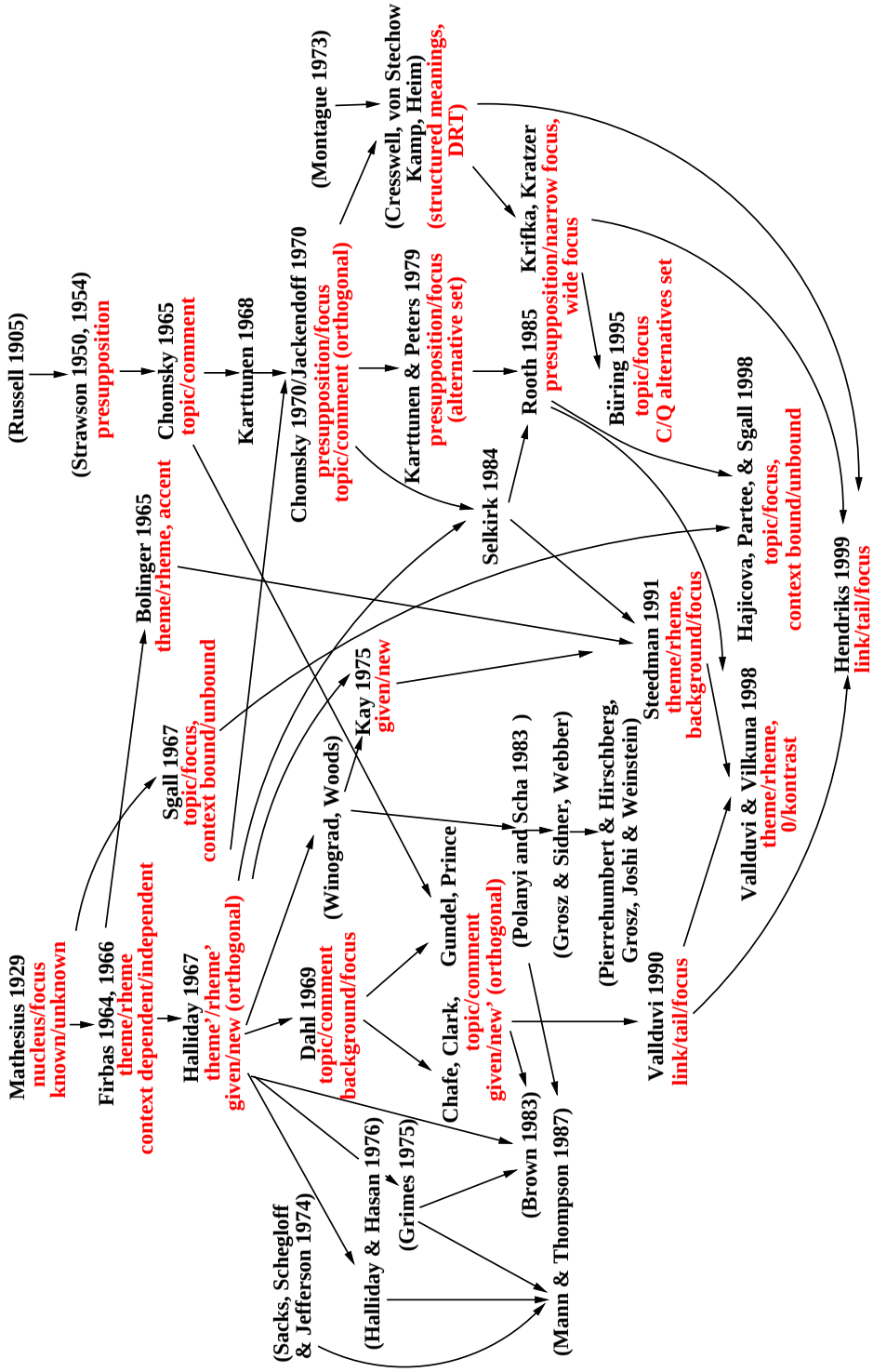
The diagram in Figure 3.1 (adapted from (Steedman and Kruijff-Korbayová, 2001)) may help to make the cross-theoretical commonalities more obvious by indicating influences and terminological dependencies in theories of Information Structure and the associated discourse semantics.

The specific approach to and terminology for Information Structure we are employing is that developed by Steedman (Steedman, 1996; Steedman, 2000a; Steedman, 2000b). This is motivated by the following considerations: Steedman's approach in a number of respects offers a synthesis of various earlier proposals, as he builds on the notion of Information Structure that originates in the work of Mathesius (Mathesius, 1975), and has been elaborated in subsequent work within the Prague School and by others (cf. Figure 3.1). It is not the particular choice of terminology that is important here, it is the insights that Steedman explicitly incorporates and the degree of their explicit formalization.

Steedman's main point is to provide a well worked out compositional analysis of English intonation in Information Structure terms. He develops a theory of grammar in which syntax, information structure and intonational prosody are integrated into one system. This is a consideration that does not play much role in our current work, but is very important from a more general theoretical as well as practical point of view, especially if one wants to address the interplay between different realization means of information structure (cf. (Kruijff, 2001) for a critical review of Steedman's work in this respect).

What is particularly important for our current enterprise is that Steedman spells out concrete correlations between information structure and intonation, that can be straightforwardly applied and experimented with. Another important point is that he is explicit

Figure 3.1: Influences and terminological dependencies in theories of Information Structure and the associated discourse semantics



about the semantics of the information structure categories he distinguishes, in terms of their discourse interpretation.

Finally, Steedman's approach to information structure in English has been used earlier to control the intonation of synthesized speech in context. Two such applications are described in (Prevost, 1995): One concerns question-answer pairs where the detailed analysis of the intonation of the question in terms of information structure is used to motivate the appropriate information structure of the corresponding answer, realized again through intonation. The other application concerns intonation in concept to speech text generation system producing short descriptions of objects, where the Theme/Rheme partitioning is motivated on text progression grounds, and the Background/Focus partitioning distinguishes between alternatives in context . The approach to assigning information structure we have developed is similar to Prevost's, but we determine the information structure partitioning on the basis of the information state representation as it dynamically evolves in various dialogue interactions.

3.2 Two Dimensions of Information Structure

Building on the findings originating in the Prague School (Firbas, 1992; Mathesius, 1975; Sgall et al., 1986), Steedman recognizes two dimensions of IS: The first dimension defines a partitioning at the utterance-level into *Theme* and *Rheme*; the second is a further partitioning of both Theme and Rheme into *Background* and *Focus*.

The Theme/Rheme partitioning as defined by Steedman reflects an *aboutness* relation, i.e., the Rheme is semantically predicated over the Theme; in terms of the question test (Sgall et al., 1986), the Theme corresponds to what the question sets up, and Rheme is what answers the question. Steedman's Theme/Rheme partitioning is similar to the partitioning into nucleus/focus (Mathesius, 1975), topic/focus (Sgall et al., 1986), theme/rheme (Firbas, 1992) and Ground/Focus (Vallduví, 1992).

The Background/Focus partitioning defined by Steedman (following Dahl) reflects an abstract notion of contrast between alternatives available/relevant in the discourse context, against which the Theme and Rheme of the actual utterance are cast. Steedman's Background/Focus dichotomy is related to Halliday's Given/New dichotomy (Halliday, 1970b; Halliday, 1985) and to the Praguian distinction between contextually bound and contextually non-bound elements. Also Vallduví's division of Ground into Link and Tail correlates with Steedman's partitioning of the Theme into Focus and Background, respectively.

Leaving more detailed analysis of the similarities and differences between the compared approaches aside, the terminologies and by an large also the underlying concepts used

Figure 3.2: Terminology Alignment

Mathesius, Daneš	Firbas,	Theme	vs.	Rheme
Sgall et al.		Topic topic proper vs. contrastive topic	vs.	Focus focus proper
Vallduví		Ground Tail vs. Link	vs.	Focus
Steedman		Theme Background vs. Focus	vs.	Rheme Background vs. Focus

by Steedman, the Prague School and Vallduví can be aligned approximately as shown in Figure 3.2.

The two dimensions of IS applying Steedman's terminology are illustrated below. (15) corresponds to the earlier (10). (16) exhibits an "inverse" Them/Rheme partitioning.[1]

(15) U: What is the status of the heaters?

S: The heater in the KITCHEN is ON.
L+H*LH% H*LL%
Background *Focus* *Background* *Focus*
Theme *Rheme*

i. θ(15.S): $\lambda x.device(heater)\&location(^{\star}kitchen)\&status(x)$
ρ(15.S): $^{\star}on$

ii. θ-AS(15.S):
$\{\exists x.device(heater)\&location(kitchen)\&status(x)$,
$\exists x.device(heater)\&location(hall)\&status(x)$,
$\exists x.device(heater)\&location(bathroom)\&status(x)\}$

iii. ρ-AS(15.S):
$\{device(heater)\&location(kitchen)\&status(on)$,
$device(heater)\&location(kitchen)\&status(off)\}$

(16) U: Which heaters are on?

[1] Henceforth, we use the ToBI ("Tones, Breaks and Indices") notation for intonation proposed in (Pierrehumbert, 1980) and (Hirschberg and Pierrehumbert, 1986). For more information about ToBI cf. *http://www.ling.ohio-state.edu/~tobi/*

S: The heater in the KITCHEN is ON.
H* L+H*LH%
Background *Focus* *Background* *Focus*
Rheme *Theme*

i. θ(16.S): $\lambda y.device(heater)\&location(y)\&status(^{\star}on)$
ρ(16.S): $^{\star}kitchen$

ii. θ-AS(16.S):
$\{\exists y.device(heater)\&location(y)\&status(on)$,
$\exists y.device(heater)\&location(y)\&status(off)\}$

iii. ρ-AS(16.S):
$\{device(heater)\&location(kitchen)\&status(on)$,
$device(heater)\&location(hall)\&status(on)$
$device(heater)\&location(bathroom)\&status(on)\}$

For each sentence, (i) provides a simplified *IS-partitioned logical form*, where θ stands for Theme and ρ for Rheme. Within Theme and Rheme, asterisks on terms (e.g., $\star carry$) indicate elements that belong to the respective Focus. These IS-partitioned logical forms represent the linguistic meaning of the sentences, and serve as input for a discourse (context) *update function*. (ii) indicates the Theme alternative set (θ-AS) and Rheme alternative set (ρ-AS), which are explained below. In the presence of Focus within Theme (indicated by a *-term), which entails contrast with a previous Theme (and hence alternatives to contrast with), the corresponding θ-AS contains more than one element. Without pitch accents in Theme, and thus without contrast, the θ-AS is a singleton set.

3.3 Semantics for IS

Elaborating on Rooth's "Alternative Semantics" (Rooth, 1985a; Rooth, 1992a), (Büring, 1995), Steedman assigns the following semantics to IS (cf. (Steedman, 2000a)):

- Theme presupposes a *Rheme-alternative set* (ρ-AS).
- Focus within Rheme restricts the ρ-AS to the singleton set corresponding to the asserted proposition.
- Theme also presupposes a *Theme-alternative set* (θ-AS).
- Focus within Theme restricts the θ-AS to the singleton set corresponding to Theme.

ρ-AS corresponds to what Rooth calls the *contextual alternative set* (Rooth, 1985b; Rooth, 1992b). θ-AS is a set of alternative themes with respect to the context, corresponding to what Rooth calls the *question alternative set* (cf. also (Büring, 1999)). The notion of alternative set is also closely related to the notion of *secondary denotation* (Karttunen and Peters, 1979).

3.4 Information Structure and Information State

In Chapter 5 we propose rules for determining information structure from the information state in GoDiS. The basis of that proposal is outlined here:

- Theme/Rheme partitioning:
  - The Theme is taken to correspond to the question under discussion a system utterance is answering.
  - Rheme corresponds to that part of the utterance that answers the question under discussion.

  For example, if the question under discussion is $?\lambda x.price(x)$, then the propositional content $price(200)$ of the answer can be partitioned as follows:
  $\theta(\lambda x.price(x))$, $rho(200)$
- Focus/Background partitioning
  - Focus marking is assigned to that part of a Theme or Rheme that distinguishes the Theme/Rheme from relevant similes in the dialogue context and/or in the domain.
  - The rest constitutes the Background.

  For example, in $class(economy)$, Focus will be assigned to $economy$, because there is a simile $class(business)$ that either was previously mentioned in the dialogue or is anyway the only suitable alternative in the domain. Thus, for example, the following partitioning of the propositional content $class(economy)\&price(200)$ can be derived when answering the question under discussion $?\lambda x.price(x)$:
  $\theta(\lambda x.class(^\star economy)\&price(x))$, $\rho(200)$

Chapter 4

Realization of Information Structure

Information structure is an inherent aspect of meaning —it is an important factor in establishing coherence with respect to context, and in getting the intended message across. In various languages, IS is realized through an interplay between intonation, word order and grammatical structure (cf. (Kruijff, 2001) for a detailed discussion from a typological perspective). Different languages combine these means in different ways. For example, in languages with a relatively fixed word order, intonation is the primary means of realizing information structure (e.g., English, Swedish). On the other hand, the so-called free word-order languages exploit primarily word order variation (e.g., Czech (Kruijff-Korbayová et al., 2002a) for a detailed discussion and more references). German provides an interesting example, because of its free word order in the so-called Mittelfeld, but fixed word otherwise. In this chapter, we discuss briefly intonation in English (Section 4.1) and word-order variation in Czech (Section 4.2) as reflexes of information structure.

Finally in Section 4.3 we turn to another aspect of realization which can also be seen as a means of realizing information structure, namely the shortening of utterances, in particular the shortening of answers to questions. We take the view that Theme and/or Background material can be left out as long as grammaticality constraints of the language at hand permit.

4.1 Realization of IS through Intonation in English

Here we concentrate on the realization by intonation in English. Steedman has argued extensively that in English Information Structure is homomorphic to Intonation Structure, and that the contour described in Pierrehumbert's (Pierrehumbert, 1980) notation

as L+H* LH% is one of the "theme tunes" that identify an intonational phrase as the theme of an utterance, while H* LL% is a "rheme tune". Within both the theme and the rheme, the presence of one or more pitch accents identifies words which contribute to distinguishing that theme and/or that rheme from other themes and rhemes that the context affords. Thus Steedman follows (Bolinger, 1965) and (Lambrecht, 1994) in viewing the role of accent in English as a single undifferentiated contrastive meaning applying to both informational components.

We adopt Steedman's account of the mapping between information structure categories and intonation, in particular different types of accents and boundary tones. While the isomorphism Steedman proposes is not entirely uncontroversial, it is the most concrete proposal we are aware of. Our other reasons for choosing of Steedman's approach are summarized in Chapter 3, p. 26.

We now show several variations of realizing the same propositional content by English "sentences" which only differ in intonation. We show the information structure partitioning corresponding to the intonation:

(17) i.

The	KITCHEN	light	is	ON
	L+H*	LH%		H*LL%
Background	*Focus*	*Background*	*Background*	*Focus*
Theme			*Rheme*	

ii.

The kitchen	LIGHT	is	ON
	L+H*LH%		H*LL%
Background	*Focus*	*Background*	*Focus*
Theme		*Rheme*	

(18) i.

The	KITCHEN	light	is	ON
	H*			LL%
Background	*Focus*	*Background*	*Background*	*Focus*
Rheme			*Theme*	

ii.

The kitchen	LIGHT	is	ON
	H*		LL%
Background	*Focus*	*Background*	*Focus*
Rheme		*Theme*	

The realizations in (17) are appropriate as answers to (19), whereas the realizations in (18i) are appropriate as answers to (20). The difference between the (i) and (ii) version in both cases concerns the alternatives to *kitchen light*: (i) is appropriate when the kitchen light is to be distinguished from lights in other locations, whereas (ii) is appropriate when the kitchen light is to be distinguished from other devices in the kitchen.

(19) What is the status of the light in the kitchen?

(20) Which device is on?

We summarize below the picture of controlling intonation according to information structure in GoDiSon the basis of the (Steedman, 2000a) proposal. More detail will be presented in subsequent Chapters 5 and 6:

- Theme and Rheme
  - Determines overall intonation pattern: Particular tunes are used to mark the Rheme-part of a sentence (predicted by the QudTR), and different tunes are used to mark the (remaining) Theme-part.
  - Theme-accents: L+H*, L*+H
  - Rheme-accents: H*, L*, H*+L, H+L*
- Focus and Background
  - Determines placement of pitch accents on particular words: Accents are assigned to the words realizing the Focus elements (predicted by ComFB or DomFB), which can appear within each Theme and Rheme.
  - Focus: marked by pitch accent
  - Background: not marked by pitch accent
- While accenting the Rheme part of a sentence is obligatory, accenting the Theme part is optional. This explains which foci predicted by the ComFB or DomFB get necessarily accented.
- Boundary tones (if any) are placed at the end of the Theme-part and/or the Rheme-part of a sentence. This gives more control over the intonation patterns than any focus-assignment rule in isolation. (Steedman's combinatory prosody takes also boundary tones into account; type of boundary tone reflects speech act, but can also relate to attitude)

4.2 Realization of IS through Word Order in Czech

For the sake of comparison, we show here the realization of examples (17) and (18i) in Czech, as an example of a language with a high degree of word order freedom, where word order variation is the predominant means of realizing information structure (Sgall et al., 1986; Hajičová and Sgall, 1987; Kruijff-Korbayová et al., 2002a).

Within the present report, this illustration serves the purpose of defending the need for information structure as a level of representation of utterance meaning abstracting away from particular realization choices. Outside the scope of the present work is the challenge that such a view of information structure presents for contemporary approaches to natural language generation (and in particular the simplified treatments common in most current dialogue systems), where the following asymmetry between handling word order and intonation exists: whereas word order (and more generally, syntactic structure) is handled in the grammar, intonation is seen as a markup assigned in a post-hoc manner. What we need instead are grammars that treat word order and intonation on a par, thus capturing their interplay (cf. (Kruijff, 2001) for a recent formally sound proposal).

Simplifying a great deal, the following holds about the ordering of words/phrases which realize arguments or modifiers of the main verb in a clause in Czech:

- Theme precedes Rheme (unmarked) or Rheme precedes Theme (marked)
- When Rheme after Theme, Rheme-Background precedes Rheme-Focus
- When Rheme before Theme, Rheme-Focus precedes Rheme-Background
- Within Theme itself, Theme-Focus precedes Theme-Background

At the same time, Rheme-Focus is the bearer of the nuclear intonation center, and Theme-Focus may but needs not be marked by an additional pitch accent.

To illustrate, we now give the Czech renditions of the earlier English examples (17) and (18i) as answers to the questions (21) and (19) (the questions are repeated here for convenience). The variation we want to point out is that in the ordering of *light in kitchen* and *turned on*. The realization in (21) is appropriate as answer to (19); the realizations in (22) are both appropriate as answers to (20) –(22a) is like in English, but (22b) places the thematic participle *turned on* at the beginning of the sentence. The English-like ordering of Rheme before Theme in (22a) is unusual in Czech. When used in a question-answers pair, it seems motivated by the resulting parallelism between the answer and the question. But the intonation pattern it requires is very marked for Czech. (For brevity, we only present the versions where the kitchen light is being distinguished from lights in other locations.)

(19) What is the status of the light in the kitchen?

(20) Which device is on?

(21)

Světlo	v KUCHYNI	je	ROZSVÍCENÉ.
light	in kitchen	is	turned on.
	L+H*LH%		H*LL%
Background	*Focus*	*Background*	*Focus*
Theme		*Rheme*	

(22) a.

Světlo	v KUCHYNI.	je	rozsvícené
light	in kitchen	is	turned on
	H*		LL%
Background	*Focus*	*Background*	*Focus*
Rheme		*Theme*	

b.

ROZSVÍCENÉ	je	světlo	v KUCHYNI.
turned on	is	light	in kitchen
L+H*LH%			H*LL%
Focus	*Background*	*Background*	*Focus*
Theme	*Rheme*		

Although it would be very interesting to investigate and compare in detail the word order and intonation variations in English and Czech, especially from the viewpoint of their contextual appropriateness and differences in meaning in dialogue, unfortunately such investigation is out of the scope of this report. The sole purpose of this brief illustration was to indicate the challenge that arises from such cross-linguistic perspective.

4.3 Information Structure and Short Utterances

In order to improve user satisfaction and efficiency of communication, what we want is a dialogue system that produces utterances that are as natural-sounding as possible. And this not only with regard to intonation, but also concerning the amount of information the utterances convey. The dialogue examples looked at so far in relation to the information structure determination rules have been of the following kind:

(23) U: How many flights are there from Helsinki to Paris on a Wednesday?
S: There are three flights from Helsinki to Paris on a Wednesday

That is, they have involved syntactically complete utterances, and the answers to wh-questions, as in the example above, have contained both a theme and a rheme. Now,

human-human dialogue is very often not like this. Repeating *from Helsinki to Paris on a Wednesday* in the answer seems redundant and inelegant, when the simple answer *Three* would do perfectly well. Indeed, looking at recorded dialogues between humans shows many instances of non-sentential utterances that avoid what is obviously redundant information. The dialogue excerpts in 24 and 25 are from the Amex travel corpus (TA is the travel agent and C the customer).

(24) TA: going where?
C: Orange County

(25) C: uh huh. and it arrives Ottawa
TA: at six ten p.m.

This section will take a look at utterances like these, and discuss how information structure and the information state can be exploited in this regard.

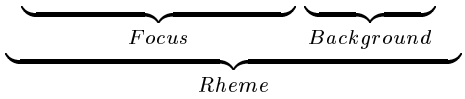
First, let us insert a terminological note. The phenomena looked at in this section are in the literature variously called ellipses, (informational) fragments, phrasal utterances, non-sentential utterances, short utterances. The term 'ellipsis' indicates that something that should be there isn't, and this is arguably only true if one uses the full sentences of written language as the norm. From a dialogue perspective, entities like *Orange County* and *at six ten p.m.* are perfectly reasonable and complete utterances in their context. We will therefore not use 'ellipsis' here, but rather the other terms, and in particular 'short utterance'.

4.3.1 Data: Answers

Looking at one kind of scenario, a wh-question immediately followed in the next turn by an answer, how much information is to be, or is, included in the answer? Thinking in terms of information structure, answers consisting of a theme and a rheme are fairly uncommon in many kinds of human-human dialogue. For instance, information-seeking dialogue like the Amex examples above contains very many short answers. A possibility is then that the answer consists of just the rheme. This seems like an obvious, and maybe the most natural, choice, as the rheme, by definition, is that which answers a question. An example would be:

(26) U: What does economy class cost?

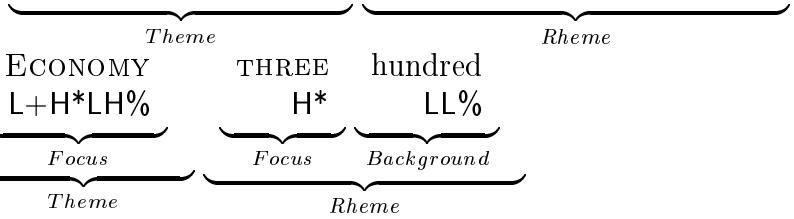
S: THREE HUNDRED Euro



The rheme may or may not be identical with the (rheme-)focus. When it's not, is it possible to produce just the focus? And is that true of the theme-focus as well if a thematic element is included in the utterance? It seems so. Consider the following, where the first part of the system's utterance, *Business class costs five hundred euro*, is a syntactically complete utterance consisting of both a theme and a rheme, and the second part, *economy three hundred*, contains a theme-focus and a rheme (here with intonational labelling to make the example clearer).

(27) U: What is the price of that alternative?

S: BUSINESS costs FIVE HUNDRED Euro
L+H* LH% H* LL%



ECONOMY THREE hundred
L+H*LH% H* LL%

For this example one may also consider the system's utterance being phrased as *Business class costs five hundred euro, economy three.* That is, with the second part of the utterance consisting of just a theme-focus and a rheme-focus.

There are of course also many examples of the rheme and the rheme-focus being one and the same, that is, there being no rheme-background. The following are taken from travel dialogues in the Göteborg Spoken Language Corpus,[1] and equivalent examples for English can be found in the Amex corpus. In 28, the customer has given part of a street address, and the travel agent now wants to know at which number on the relevant street the customer lives.[2] Of course, the distinction between the material included in the rheme

[1] `http://www.ling.gu.se/projekt/SLSA/SLcorpus.html`

[2] In this section, the transcription conventions – although sometimes simplified to leave out information that is not relevant to our discussion – of the original transcriptions are kept. For example, the Amex transcriptions include capital letters, question marks and full stops, whereas the Göteborg transcriptions only use lower case and no punctuation marks in order not to impose (too many) written-language conventions on spoken language.

and that included in the rheme-focus depends on the granularity of one's semantics. But assuming that the travel agent's question in 28 is $?\lambda x.street\_number(x)$ and the customer's answer *seven*, and in 29 $?\lambda x.home\_phone\_number(x)$ and 53411, we find that rhemes and rheme-foci coincide.

(28) TA: nummer
number
C: sju
seven

(29) TA: telefonnummer hem
home phone number
C: fem tre fyra ett ett
five three four one one

The answers above have been short in both a syntactic and a semantic sense: the answers aren't complete sentences, and the intepretation of them doesn't yield a full proposition. Such utterances cause problems for many computational approaches as standard tools for parsing are typically made for complete sentences. In this context it is also interesting to note the existence of answers that are syntactically complete, but semantically short – in relation to a particular domain. Above we saw an example of a syntactically and semantically short answer when a travel agent asked *home phone number* and got the reply *five three four one one*. A few turns later in the same dialogue we find the following (a forward slash indicates a short pause and curly brackets are used to add in material that isn't pronounced in the dialogue but makes the reading of the transcription easier):

(30) TA: då ska vi se här / dagtid mobil arbete
let's see / daytime mobile phone work
C: {j}a ö{h} ta då e{h} ja noll tretti{o}ett fyrti{o}tre nitton nitton
well eh then take eh zero thirty-one forty-three nineteen nineteen[3]

Here, the user answer contains a few disfluencies, or own communication management (OCM) devices, but ignoring these we find a (syntactically complete) imperative sentence, *(Then) Take zero thirty-one forty-three nineteen nineteen*. What answers the travel agent's question is just the phone number, *zero thirty-one forty-three nineteen nineteen*, and the other material in the utterance – the OCMs and the imperative verb – is not

[3]In the original transcription, the phone number is even said across two turns, with the travel agent producing a backchannel *yes* in-between.

thematic. But, semantically, in relation to the domain, in this case a question like $?\lambda x.daytime\_contact\_no(x)$, the utterance is in some sense short, in that 031431919 is the rhematic information of the customer's utterance in relation to the preceding question (and there is no thematic information, in relation to the preceding question).

Thus, the user utterance can be seen as a rheme together with some other material. This other material may provide other information that is relevant to the domain (e.g. if the customer had continued his utterance by *and address...*) or it may not. It may also be material that is due to language production factors, such as much of the utterance above which seems to be due to the customer talking while recalling, or maybe choosing, the phone number. Look also at the two question-answer sequences in 31, taken from an informal conversation in the Göteborg Spoken Language Corpus.

(31)
A1: va{d} e0 de{t} som e0 i kakao i i vänta nu i te /// å0 å0 kaffe så e0 de{t}
what is it that's in cocoa in in hang on in tea /// and and coffee there is
B1: koffein
caffeine
A2: koffein men va{d} e0 de{t} i kakao
caffeine but what's in cocoa
B2: de{t} e0 väl koffein också
I guess that's caffeine too

The question in A1 can be represented by something like $?\lambda x.substance\_in(tea, x)$ & $substance\_in(coffee, x)$, and the question in A2 by $?\lambda x.substance\_in(cocoa, x)$. Answer B1 is syntactically and semantically short. Answer B2, on the other hand, is syntactically complete. But informationally there is something very similar between the B1 and B2 answers. Both contain a rheme, $caffeine$, and then B2 includes extra material, indicating degree of certainty, which may or may not be dealt with in the domain.

4.3.2 Data: Questions

As could be noted already in the examples in the previous section on short answers, questions can also be short. In a travel agency domain, the full question *Where are you going?* isn't needed; *Going where?* suffices perfectly. As does *Home phone number?* instead of *What is your home phone number?*. So it seems reasonable to assume that short utterances are important for all sentence types.

Another possibility is that questions influence answers, in terms of whether they are short or not, and if short, how short. Take the following two (constructed) dialogues:

(32) U1: What does economy class cost?
S1: (Economy class costs) five hundred euro
U2: What does business class cost?

(33) U1: What does economy class cost?
S1: (Economy class costs) five hundred euro
U2: And business?

It may be possible for S2 in both of these dialogues to be either *three hundred euro* (rheme only) or *three hundred* (rheme-focus and part of the background), but the focus-only answer, *three*, if it works at all for this example in this domain, may be slightly more felicitous in the second dialogue than in the first, and a full theme-rheme answer may seem slightly less felicitous in the second dialogue than in the first. A corpus study or user tests with an implemented system would be needed to test and further explore these ideas, neither of which has been done here.

However, an interesting example of the interplay between questions and answers in relation to thematic and rhematic information and how much of each is produced, can be found in the following excerpt from a HCRC Map Task dialogue (F for follower and G for giver):[4]

(34) G1: where are you in relation to the top of the page just now?
F1: Uh, about four inches.
G2: Four inches?
F2: Yeah.
G3: Where are you from the left-hand side?
F3: About two.

G1 is a sentential question, and F1 is a short answer to G1. G2 is a short clarification question following upon the short answer in F1. G3 is again a sentential question, and very similar to G1, but compared to that question, the temporal indication, *just now*, has been left out (note also that in G3, a short question like *And from the left-hand side?* seems to work just as well). F3, finally, is a short answer to G3.

Here the domain, the task at hand, is likely to have an influence on both questions and answers. In particular, when we get to F3, the measuring unit, inches, is no longer part of the utterance. This was rhematic information in F1, and was again repeated in G2, so that in F3 it no longer needs to be repeated. The follower assumes that the giver is able to interpret *about two* as referring to inches.

[4]The example is taken from `http://www.hcrc.ed.ac.uk/dialogue/maptask.html`

4.3.3 Short utterances using the information state

Clearly, the short utterances looked at above are complete in their context in the dialogue since participants are perfectly, bar misunderstandings, able to recover their full meaning. An answer immediately following a question doesn't need to repeat thematic material, which means that this material must be readily available to the dialogue participants. Our job is to, for interpretation, connect user short utterances with the appropriate context to get a full interpretation in a way analogous to what human dialogue participants do, and, for generation, produce utterances that take into account the information already in the information state and give an appropriate amount of thematic and rhematic information.

How would short utterances be accounted for in a dialogue system like GoDiS? This deliverable contains no implementation of short utterances, but let us attempt a few theoretical ideas.

Ginzburg, in e.g. (Ginzburg, 1999), discusses questions maximal in QUD as being available for ellipsis resolution for a subsequent short utterance. Reformulating his situation theory approach as something closer to what's used in GoDiS, Ginzburg provides the following example:

(35) A: Who will drive the train?
Question expressed (i.e. on QUD): $?\lambda x.will\_drive\_the\_train(x)$
B: Bill
Content of answer phrase: b
Content of short answer: $will\_drive\_the\_train(b)$

The content of the short answer is clearly derived from the answer phrase and the question on QUD, here by straightforward functional application.[5] Previous GoDiS versions have incorporated this treatment of short utterances, so that a short utterance made by a user can be fully interpreted given a suitable question topmost on QUD (the question being either explicitly asked, or put on QUD through accommodation), see e.g. (Larsson, 2002).

However, in the current work we are concerned with system output, hence with the system's ability to *produce* short utterances. More specifically, from an abstract specification, marked for information structure, of an utterance to be produced, together with any other relevant information in the information state, we want the system to be able to decide how

[5](Ginzburg, 1999) goes on to discuss syntactic and semantic mismathes between the constructed content of the short answer and its intended meaning, draws the conclusion that neither purely syntactic nor purely semantic approaches to ellipsis resolution give the desired result, and instead formulates syntactic restrictions on possible short utterances, the restrictions being associated with the questions.

much of the theme and the rheme is to be included in the realisation. The system will need to decide when to generate the full theme and rheme, when just the rheme, when just the (theme and/or rheme) foci. The key to this decision is: the material 'left out' must be readily recoverable from the context.

Take the example of the Map Task dialogue above. In F3, the follower doesn't need to mention 'inches', which *was* included in the answer to the very similar question in G1. It is noteworthy that for all other measurements in the dialogue, the word 'inches' *is* included. This may suggest that something that has been mentioned in a dialogue need not remain thematic throughout the dialogue; there may be a local context of say a few turns that establishes and maintains something as thematic. This needs to be further explored before a theory of the production of short utterances can be formulated.

Another aspect that may licence the leaving out of rhematic material is domain knowledge. For example, in the domain of house buying, people in the know may cite a price in the form of e.g. '995', with the intended meaning of '995 *thousand (Swedish) kronor*'.[6]

There may not be any strict criteria for what does work in a given situation, e.g. theme-and-rheme and just-rheme may both work, or full rheme and rheme-focus with some of the background. But, it may be possible to establish criteria for what works *less well* than other alternatives. For instance, a focus-only answer may be very hard to interpret in a certain context, and in another context a full theme-rheme answer may be hopelessly redundant.

(Fernández and Ginzburg, 2002) proposes a taxonomy of short queries and short answers. The information structure of questions hasn't been dealt with in any depth in this deliverable. That is something that needs to be done if information structure is to be used to produce short questions. A reasonable assumption is that the amount of thematic and rhematic information in a question depends on locallly established themes and domain knowledge in a way similar to answers.

As for possibly sentential answers consisting of a rheme, or part thereof, together with non-thematic material (in view of the preceding question), it can be noted that some of these bear a resemblance to what Valluví refers to as *informational fragments*. In his talk at ESSLLI in 2001, he discussed examples like the ones in 36 and 37 (where small capitals are use to mark intonation centre). Valluví argues that although B's answer in 36 is sentential, it is as much of an informational fragment as B's utterance in 37. The extra material in 36, the pronouns, is just there for morphosyntactic reasons.

(36) A: How does he feel about Bill?
B: He LOVES him.

[6]The example is due to Robin Cooper.

(37) A: Who does John love?
B: BILL.

In the corpus examples above, sententials consisting of a rheme together with non-thematic material, the extra material was not there for morphosyntactic reasons, but seemed rather to have something to do with speech production factors and to add information regarding degree of certainty for the rhematic material. However, in view of a dialogue system and a particular domain, this extra material may not necessarily contribute something to be retained, and in view of the preceding question, the answer may therefore be seen as some form of a short answer. For generation, we'd want a system that produces utterances consisting of a rheme together with non-thematic material that is otherwise informative.

4.3.4 Human-human dialogues vs. human-computer dialogues

As a final point, let us briefly consider the following question: do we want our dialogue system to behave just like humans when it comes to short utterances? Are human-human and human-computer dialogues equivalent? The corpus examples above were all from human-human dialogue; can we model our system's behaviour directly on these?

For interpretation, whenever a user produces a short utterance, the system should obviously and ideally be able to make sense of it (at least to the extent that a human would be able to). Of dialogue system development relevance here, is an issue such as how long questions stay under discussion. If a user produces a short utterance, what question could it be related to? Here, the study of human-human dialogue may have a role to play, in that we want a system to be enable human users to behave as naturally as possible.

When it comes to generation, a dialogue system shouldn't necessarily behave like a human. Speech recognition problems risk creating a vast amount of misunderstanding and non-understanding, and many people feel they can't trust a system as well as a human being. One way is then to incorporate some form of user modelling; when talking to naive users the system could produce more full theme and rheme utterances, whereas experienced users would be met by more rheme-only and focus-only utterances. Recognition confidence level could also be used; a high confidence level means a short utterance can be produced, and a low level results in full theme and rheme utterances. Or, if the user has many short utterances, then the system should be able to adapt and also use short utterances. The aim, after all, is to produce a system that is user friendly, that conveys reliability (the user can be confident that the system has understood) and yet avoids cumbersome utterances with unnecessary information. So, for generation, a corpus study is relevant to get an idea of how humans communicate, and this can then be modified in varous ways in a dialogue system context.

4.4 Summary

Information structure is an inherent aspect of utterance meaning, which reflects a partitioning of meaning according to how an utterance relates to the context and how it updates it. At the same time, information structure is a level of meaning which unifies a range of interacting contextually-dependent aspects of utterance realization. In this chapter, we addressed the realization of information structure trough intonation in English and briefly alluded at its realization though word order in Czech. We also discussed short utterances from the viewpoint of information structure and suggested that the same underlying abstract representation of Themes, Rhemes and Foci also be used to control utterance shortening.

We pointed out that the task of capturing the interacting means of information structure realization in an integrated fashion in natural language generation (or parsing) is a challenge for contemporary systems. In the remainder of this report we concentrate on the realization of information structure through intonation, because our primary goal in the present work has been to improve the results of synthesized spoken output of a dialogue system. Taking into consideration other means of realization simultaneously remains as a challenge for future development.

Chapter 5

Information Structure Determination from the Information State

In this chapter, we discuss how information structure can be determined from the information state which represents the context in a dialogue system. We concentrate on determining the information structure partitioning, whereas the corresponding realization was discussed in Chapter 4. To begin with, we briefly recapitulate the notion of information state in the GoDiS system. Then we present our approach to assigning information-structure partitioning on the basis of this information state. The current approach builds on and extends previous work in the TRINDI project (Engdahl et al., 2000).

5.1 The GoDiS Information State

GoDiS, the experimental system we have been developing, uses the Information State (IS) approach to dialogue modeling pursued in the SIRIDUS project. The type of record assumed for the GoDiS information state is a version of Ginzburg's *dialogue gameboard*, (Ginzburg, 1996). Slightly different information states have been used at different stages in the development of GoDiS as part of various projects and applications. For the discussion in this chapter we assume the 'minimal' information state in Figure 5.1.[1] Each dialogue participant has her own information state, although only the system's IS is modeled in GoDiS.

The information state is divided into a PRIVATE and a SHARED part, the latter containing information that the agent assumes to be shared by the participants in the dialogue. In

[1] The basic version of GoDiS as used in the Siridus project is described in (Larsson et al., 2002).

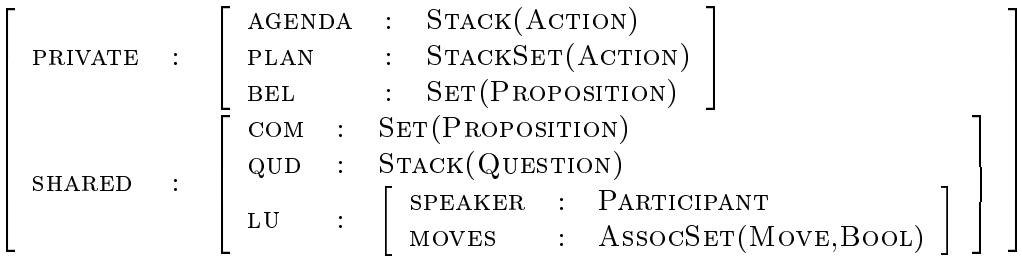


Figure 5.1: The Information State in GoDiS

addition to information about the latest utterance (speaker and move(s)), the SHARED part contains shared commitments – a set of propositions – as well as the QUD, a stack of questions under discussion.[2]

When one of the participants asks a question, it is pushed onto the QUD, and it is popped off when it is answered. When the user supplies information not asked for by the system, but which is relevant to some question in the domain, *question accommodation* occurs, that is, the system pushes the relevant question onto QUD in order to be able to integrate the information given by the user.

Current versions of GoDiS make use of keyword and keyphrase spotting for interpretation of the user's input, and a simplified semantics. For instance, a user utterance like "I'd like to go to London" will be recognized as a move giving a destination through the word 'to' followed by a city, and its contents will be represented in the shared commitments as the predicate-logic proposition $dest(london)$. The corresponding question, where does the user want to go, will be represented on the QUD as $?\lambda x.dest(x)$.

In the PRIVATE part, the plan contains the system's long-term goals, i.e. a list of actions to carry out, whereas the agenda contains more immediate actions, typically taken from the plan. The actions, both the long-term ones and the more immediate ones, can be either communicative actions, such as ask the user a question, or non-communicative actions, such as check whether a device in the home is on or off. In the remainder of this chapter we will ignore the non-communicative actions as only the communicative ones concern us here.

Let us go through an example to look at the dynamics of the agenda, plan, and QUD. Consider the following dialogue in the home device domain:

(38) S1: Welcome to the home device manager!
U1: hello
S2: What can I do for you?

[2]Ginzburg's QUD is a set partially ordered by 'conversational precedence', see e.g. (Ginzburg, forthc).

U2: switch on the light
S3: Where?
U3: in the kitchen
S4: The lamp in the kitchen is now switched on

In U2, when the user tells the system to switch on the light, the system is able, from the user utterance, to identify the task the user is interested in – in this case doing something to a particular device. In the domain, this task is connected to a particular sequence of actions. The system takes this sequence of actions, and pushes them onto the plan in the private part of the information state.

The system then observes that one of the actions in the plan, to ask the user what she wishes to do to some device, need not be carried out because the user has already given the answer, namely to switch something on. This means that the system deletes the action $findout(\lambda x.action(x))$ (where *action* is not a plan type action, but rather an action that the user wishes the system to carry out, such as switch on, switch off, dim, check the status), and accommodates the corresponding question $?\lambda x.action(x)$ on the QUD. This enables the system to integrate the user answer $(answer(action(switch\_on)))$, which involves popping the QUD, and adding the resulting proposition $(action(switch\_on))$ to the commitments in the shared part of the IS.

The same is done for the device type, which the system has an action to ask for, and which the user has already given an aswer to, *a light*.

The next thing on the system's agenda is the action $findout(\lambda x.location(x))$. The user has supplied no information concerning this, so the system pushes this action onto its agenda. All of the above – accommodating a task and questions, moving actions from the plan to the agenda, managing the QUD, etc – has been done with the help of specialised update rules, for updating the information state. The next thing the user does, choosing its next move based on what's on top of the agenda, is done with the help of a selection rule. The next move is simply $ask(\lambda x.location(x))$ (so the action $findout$ is converted into the move ask by the selection rule). The generation module then uses the lexicon resource, which contains pairs of moves and strings, to find the output string corresponding to the next move $ask(\lambda x.location(x))$, and with the help of the output module puts forward the utterance in S3, *Where?*.

After the system question has been asked, it is grounded and integrated, the latter meaning that the question $?\lambda x.location(x)$ is pushed onto QUD.[3] That is, before the system asks a

[3]This is an optimistic system, which assumes that the user always hears and understands system utterances. A cautious system would wait until the user indicated understanding, e.g. by giving explicit feedback, before pushing the question onto QUD.

question, (the action to ask) the question is typically on the agenda, and it is only after it has been asked that is ends up on QUD. If the system addresses a question asked by the user, the system will have pushed the question onto QUD as part of the integration of the user utterance, that is, the question will be on QUD before the system selects its next move and produces the corresponding utterance.

5.2 Theme/Rheme Assignment Using QUD

As we discussed in Chapter 3, utterances can be seen as divided into a part which relates them to the context, and an informative part, which specifies the information that should be added to the hearer's information state. Ginzburg (Ginzburg, 1999) uses the terms *ground* and *focus* for these two parts, and formulates the following:

> Every utterance-type can be partitioned into two compontents (not necessarily syntactic constituents) one of which constitutes the ground, the other the focus. Every utterance-type contains a focus, though some utterances might contain only a focus[4]

Ginzburg then describes the relationship between focus-ground (f/g) and QUD, and describes it as a felicity condition:

> An utterance with a given f/g partition requires for its felicity the maximality in QUD of a certain question, one whose defining property is identical with the scope generated by the focus constituent(s)[5]

He illustrates this with the two examples below: (39a) presupposes QUD-maximality of the question (39b), whereas (40a) presupposes QUD-maximality of the question (40b).

(39) a. [JILL]$_{FOCUS}$ [likes Bill]$_{GROUND}$
b. *who likes Bill*

(40) a. [Jill likes]$_{GROUND}$ [BILL]$_{FOCUS}$
b. *who does Jill like*

[4]Focus-only utterances, or short utterances, will be discussed in 4.3. The current chapter will only be concerned with sentential utterances.

[5]A question being maximal in QUD corresponds to a question being topmost on QUD in the GoDiS information state.

(Engdahl et al., 2000) proposed to capture Ginzburg's felicity condition by the *Focal Question Presupposition* rule:

> **Focal Question Presupposition (FQP):** If an utterance u has narrow focus over x, u (focally) presupposes a question q obtained by abstracting x over (the content of) u

If the presupposed question is *not* topmost on QUD, a *Focal Question Accommodation* rule puts it there. For generation, (Poesio et al., 2000) formulated the *QUD-based Focus Assignment* rule to enable the system to assign focus to its own utterances:

> **QUD-based Focus Assignment (QFA):** If there is a question q topmost on QUD, and an utterance u with content c is to be uttered, and q is obtained by abstracting component f over c, then c should focally presuppose q (i.e. focal stress should be put on the part of u that corresponds to f).

That is, reusing Ginzburg's examples above, if the question *who does Jill like* (i.e. $?\lambda x.like(j,x)$) is topmost on QUD, and the system is to utter *Jill likes Bill*, then focal stress is assigned to *Bill*.

So far the background. We are now going to go on to integrating these QUD-based rules together with other rules (in section 5.3) into one framework, which will allow their applications to be combined and their areas of application to be extended. Before we discuss our reformulation of the QFA rule and its application in more detail, let us first make a terminological adjustment.

Ginzburg's terminology is based on Vallduví's *information packaging*. Based on our alignment of different information structure terminologies in Chapter 3, his notion of focus is thus comparable with the Praguian notion of Focus (Sgall et al., 1986), and with Steedman's notion of Rheme (Steedman, 2000a). We will therefore refer to the part of an utterance which is informative with respect to a given QUD, as the Rheme.

(41) U: How much is the flight?
QUD established by question: $?\lambda x.price(x)$
S: $\underbrace{\text{The price is}}_{Theme}$ $\underbrace{\text{423 EURO}}_{Rheme}$.
Proposition of answer: $price(423\_Euro)$
Theme-part of answer: $\lambda x.price(x)$
Rheme-part of answer: 423_Euro

5.2.1 QUD-based Theme/Rheme Determination

Substituting the term Rheme for focus, and moving to the Theme/Rheme framework, the QUD-based Focus Assignment rule of (Poesio et al., 2000) can now be seen as dividing an utterance into a Rheme and a Theme. QFA will therefore be replaced by the following rule:

> **QUD-based Theme-Rheme determination (QudTR):** If there is a question q topmost on QUD, and an utterance u with content c is to be uttered, where q is obtained by λ-abstracting over c, then that part of c which corresponds to q belongs to the Theme of u, and the other part of c is the informative part which constitutes the Rheme of u.

As a simple example, consider again the following:

(41′) U: How much is the flight?
S: $\underbrace{\text{The price is}}_{Theme}$ $\underbrace{\text{423 Euro}}_{Rheme}$.

QUD before answer: $?\lambda x.price(x)$
Proposition of answer: $price(423\_Euro)$
Theme-part of answer: $\lambda x.price(x)$
Rheme-part of answer: 423_Euro

The user utterance pushes the question $?\lambda x.price(x)$ onto QUD. The contents of the system's utterance in the next turn, $price(423\_Euro)$ needs to be partitioned as an answer to this question, i.e., the informative part of the system's utterance corresponds to 423_Euro.

The following example serves to illustrate how the same propositional content can be partitioned differently depending on the preceding dialogue context, which is reflected in the QUD. In (42) *fly* is marked as rheme because the question on QUD concerns *how* the travelling can be done, whereas in (43) *to Frankfurt* will be the rheme because the question is about a destination, and 44 concerns places of departure so *from Saarbrücken* is the rheme.[6]

[6]In 43 and 44 one may argue that the prepositions, *to* and *from*, shouldn't be part of the rheme but rather of the theme. We take information structure partitionings to be at the level of semantics, and there, indeed, we have *dest* and *dep* (the semantic elements corresponding to the words *to* and *from*,respectively) as thematic. We see the issue of just which words in the realisation are to belong to which information unit – to the theme or the rheme – as in many ways a separate issue, and here choose to adhere to traditional syntactic structure.

(42) U: How can I get from Saarbrücken to Frankfurt?
S: It is possible to fly from Saarbrücken to Frankfurt.
QUD: $?\lambda x.how(x)$
Proposition: $how(fly)\&dep(sb)\&dest(fra)$
Theme-part: $dest(fra)\&dep(sb)\&\lambda x.how(x)$
Rheme-part: fly
$\underbrace{\text{It is possible}}_{Theme}\ \underbrace{\text{to fly}}_{Rheme}\ \underbrace{\text{from Saarbrücken to Frankfurt.}}_{Theme}$

(43) U: Where can I fly from Saarbrücken?
S: It is possible to fly from Saarbrücken to Frankfurt.
QUD: $?\lambda x.dest(x)$
Proposition: $how(fly)\&dep(sb)\&dest(fra)$
Theme-part: $how(fly)\&dep(sb)\&\lambda x.dest(x)$
Rheme-part: fra
$\underbrace{\text{It is possible to fly from Saarbrücken}}_{Theme}\ \underbrace{\text{to Frankfurt.}}_{Rheme}$

(44) U: From where can I fly to Frankfurt?
S: It is possible to fly from Saarbrücken to Frankfurt.
QUD: $?\lambda x.dep(x)$
Proposition: $how(fly)\&dep(sb)\&dest(fra)$
Theme-part: $how(fly)\&dest(fra)\&\lambda y.dep(y)$
Rheme-part: sb
$\underbrace{\text{It is possible to fly}}_{Theme}\ \underbrace{\text{from Saarbrücken}}_{Rheme}\ \underbrace{\text{to Frankfurt.}}_{Theme}$

5.2.2 QUD-less Utterances in GoDiS

Every time a user or the system asks a question that is successfully integrated, the question will end up on QUD right after the turn during which it was asked, and before (the planning of) the utterance in the next turn.[7] This means that when the system is going to select and produce a response to a user question, in the subsequent system turn, the question is available on QUD, and when the system is selecting and producing a question move, the action to ask the question is on the agenda, but not on QUD, as we saw above.

[7]Again, assuming a system that uses an optimistic approach to grounding and integration

In theory, if the system is to provide an answer to a question, the question must either be on QUD, or the system has some action topmost on its agenda which says that such an answer should be provided (the latter making the user do question accommodation). However, in current implementations of the GoDiS system there are utterances which are not seen as answers to questions. How is a theme-rheme partition assigned to such utterances? Examples are greetings, certain questions, and the contents of some $inform$ moves.

For an illustration, let us return again to the home domain dialogue above, which is repeated below as 45.

(45) S1: Welcome to the home device manager!
U1: hello
S2: What can I do for you?
U2: switch on the light
S3: Where?
U3: in the kitchen
S4: The lamp in the kitchen is now switched on

The utterance in S1 is a greeting. This move is selected by the system at the very start of the dialogue, and its production and integration does not involve the QUD in any way. The question in S2 is $\lambda x.task(x)$, and although its integration, after it has been asked, ivolves the QUD, there is no presupposed question on QUD before it's asked that can help the determination of its theme-rheme structure. Similarly, for the utterance in S4, which is the content of an inform move, before it is uttered there is no question on QUD, but only the agenda action to inform the user, and after it has been uttered nothing is pushed on QUD.

One way of handling these 'QUD-less' utterances, would be to change the theory, and consequently the implementation, such that all utterances are seen as answers to questions, even questions themselves. Greetings could be seen as addressing some abstract issue such as what to do next, and questions what question to ask the user next. This would presumable make the whole greeting and the whole question rhematic. Inform moves could be treated similarly, as addressing the issue of what to inform the user.

Another possibility is quite simply not to assign any information structure to these utterances, and let an independent intonation module assign pronunications.

5.3 Focus/Background Assignment Using Parallelism

We have formulated rules for determining the Theme-Rheme partition of an utterance based on the information state. Next, we need to write rules for determining the Focus-Background within each of the Theme and Rheme. This will be done using the notion of (semantic) parallelism, which we define as follows (we take an information unit to be a basic term, a Theme or a Rheme, or a proposition without Theme-Rheme partitioning):[8]

> Two information units
>
> $a = a1 \circ a2$
> $b = b1 \circ b2$
>
> where $\circ$ means composition, are parallel when $a1$ is parallel with $b1$ and $a2$ is parallel with $b2$
>
> Two basic terms, that is, further undecomposable information units, are parallel when they are either identical or alternatives.

We further define 'identical' elements of a sort and 'alternatives' belonging to the same sort but being non-identical.[9]

Having introduced the idea of parallelism, we will now go on to defining two different and complementary rules for determining Focus-Background that make use of this notion. The difference between the two rules lies in what the source of alternatives (and identicals) is taken to be; the first rule, ComFB, uses the discourse context, whereas DomFB uses the domain.

[8](Huet, 1975) describes the idea of using higher-order unification to solve equations involving the λ-calculus, such as for decomposable information units in our definition of parallelism in this section. Parallelism and higher-order unification have later been used in e.g. (Dalrymple et al., 1991), (Pulman, 1997), (Gardent and Kohlhase, 1997).

[9]The semantic sorts used in GoDiS are of a non-hierarchical kind, which means that the only distinctions that are needed is whether two basic terms belong to the same sort, and if they do, whether they are identical. Gardent and Kohlhase, (Gardent and Kohlhase, 1997), use a hierarchy of sorts, and have a three-way distinction paraphrasable as: 'Two elements are *similar* if they have a common sort, *contrastive* if they have a distinguishing sort, and *parallel* iff they are both'. Should the need for a more complex hierarchy of sorts in GoDiS arise, something similar could be used there.

5.3.1 Using Shared Commitments: The ComFB Rule

In (Pulman, 1997), higher-order unification is used to account for the focus-background partition of sentences that can be construed as being parallel (in our sense above) to some other sentence. His ideas were used in (Engdahl et al., 2000) to formulate the following rule:

> **Focal commitment presupposition (FCP):** if an utterance u with content s can be structured into a background b and a focus f, u presupposes a proposition c which can be structured into components p and a such that p and b, and a and f, are parallel

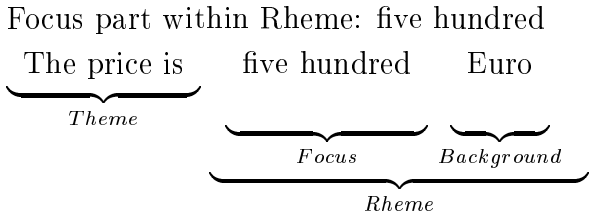
Given our definition of parallelism above, we are now in a position to state our first rule for *assigning* Focus-Background within our information structure framework:

> **Commitments-based Focus-Background determination (ComFB):** Assuming parallelism has been determined according to its definition, assign focus to any element in an information unit being an alternative to an element in a parallel information unit.

This rule then expresses the observation that the placement of intonation center is influenced by the contents of the shared commitments (the propositions that dialogue participants (believe they) have jointly committed to in the dialogue).

As a slightly more complex example than the QudTR examples above, take the following dialogue fragment where the system and user are in the middle of discussing a particular destination (note that here we're using a more fine-grained representation of prices as compared to the examples above):

(46) U1: How much is the business flight?
S1: The price is one thousand Euro
Among shared commitments: $price(1000, euro)$
U2: And how much is the economy flight?
QUD established by question: $?\lambda x.\lambda y.price(x, y)$
Answer to be generated: $price(500, euro)$
S2: The price is five hundred Euro
$\underbrace{\text{The price is}}_{Theme}\ \underbrace{\text{five hundred euro}}_{Rheme}$



When the system utterance in S2 is to be generated, the QudTR rule is first applied, with the result that (the elements in the utterance corresponding to) 500 and *euro* constitute the Rheme, and (the elements in the utterance corresponding to) $\lambda x.\lambda y.price(x, y)$ are the Theme. As 1000 is among the shared commitments, and assuming a domain where numbers are alternatives of each other, 500 will then be the focus within the rheme by the ComFBrule. The other element in the Rheme, *euro*, will not be assigned focus as it is identical with *euro* in the commitments and not an alternative. Hence, *euro* will be part of the (rheme-)background.

The element *price* may or may not receive (theme-)focus, depending on what sorts are chosen to belong to the domain. Theme-foci will be further discussed in section 5.5.2.

5.3.2 Using Domain Knowledge: The DomFB rule

We will now turn to our second rule for determining Focus-Background. The basic idea here is to use what we know about what entities are available in the *domain* to assign focus to what distinguishes among them. We follow up on Prevost's work (Prevost and Steedman, 1994; Prevost, 1995).

We define the rule as follows, and it too makes use of parallelism:

> **Domain-based Focus-Background determination (DomFB):** Assuming parallelism has been determined according to its definition, assign focus to any element in an informativity unit having an alternative in the domain

Consider the following example:

(47) U: Is there a flight from Saarbrücken to Frankfurt on Monday?
S: There is a business flight
QUD: $?(how(fly)\&dep(sb)\&dest(fra)\&dep\_day(mo))$
Proposition: $class(business)$

Theme-part: no theme
Rheme-part: *class*(*business*)
$\underbrace{\text{There is a business flight}}_{Rheme}$
Focus part within Rheme: *business*
$\underbrace{\underbrace{\text{There is a}}_{Background}\ \underbrace{\text{business}}_{Focus}\ \underbrace{\text{flight}}_{Background}}_{Rheme}$

Assuming that the QudTR rule has already been applied, focus will be assigned to *business* within the rheme, given that in the travel agency domain in question, *business* has the alternative *economy*.

The DomFB thus motivates producing for example ECONOMY *class* rather than *economy* CLASS no matter whether any class has been already mentioned or not, simply because *economy* is what distinguishes *economy class* from *business class* in the domain; and there is nothing else in the (travel) domain which would be described as *economy x*.

On the other hand, in the house domain, we want to be able to control the choice between KITCHEN *light* and *kitchen* LIGHT, depending not only on the domain (or the situation in a given house) but also on the particular dialogue context. This means we need to allow the ComFB and DomFB rules to interplay. In particular, if ComFB assigns some Focus, DomFB is not applied, because we do not want it to overwrite that assignment or assign more Foci.

The application of the DomFBrule needs to be constrained, however, because under the definition we gave so far, just about everything can be construed as an alternative to something else. One way is by taking great care when defining the semantic sorts of a particular domain, so that only those elements that can be assigned focus in the domain are assigned to a semantic sort. However, this is not a very modular way of doing things, and it may still lead to overgeneration of foci. Further work is clearly needed in this area.

5.4 Accommodation

The felicity condition, **QudTR**, states the connection between a particular Theme/Rheme partitioning and a particular question topmost on QUD. If the presupposed question is *not* on QUD, it can be placed there using question accommodation. We formulate the following rule:

Theme-Rheme Accommodation on QUD (TRAcc): If the content of an utterance has a certain Theme-Rheme partition, and abstracting over the Rheme gives a question that is not topmost on QUD, make that question topmost on QUD

This rule can be used in interpretation. A simple example is the following:

(48) U: $\underbrace{\text{I would like to go}}_{Theme}$ $\underbrace{\underbrace{\text{to}}_{Background}\ \underbrace{\text{MILAN}}_{Focus}}_{Rheme}$ H*

Proposition: $dest(mil)$

Rheme-Focus: mil

Presupposed question: $\lambda x.dest(x)$

We haven't said anything about the recognition of intonation centres, the subsequent determination of Foci (and Backgrounds) and Rhemes (and Themes), and the mapping of information structure from a string to a semantic representation. But, assuming we have some way of doing this, the question that is the result of abstracting over the information-structurally marked content of the user utterance, can be accommodated on QUD.

Of course, there are a number of problems here. Intonation patterns may be ambiguous between broad and narrow focus, which means that it may difficult to establish the rheme, and hence the theme of the utterance. The example above is a simple one, but it is not very difficult to imagine more complex ones where several different questions may arise depending on what one abstracts over, and all of the questions may be relevant to the domain. In that case, it is possible that one would need complex reasoning or dialogue history in order to establish which the presupposed question is.

QUD accommodation is relevant also in generation. For example, it can be exploited in the generation of helpful, "overinformative" or indirect answers. Compare the following two dialogues:

(49) U: I'd like to fly to Milan TOMORROW

S: I'm sorry, Milan's all booked out tomorrow. Could you go tonight or the day after tomorrow?

(50) U: I'd like to FLY to Milan tomorrow

S: I'm sorry, there are no more flights to Milan. Do you want to take the train?

Assuming narrow focus, in the first dialogue, the system accommodates the question $?\lambda x.dep\_day(x)$, finds alternative ways of answering the question when the user-supplied answer didn't work, and poses an alternative question to the user. Again assuming narrow focus, in the second dialogue the presupposed question is instead $?\lambda x.how(x)$.

And in one of the dialogues above, repeated below as 51, the system wants to give the user more information than he asked for, namely that there is a *business* flight, and in order to do so and to assign the appropriate information structure to that utterance, the system in some sense accommodates the question $?\lambda x.class(x)$.

(51) U: Is there a flight from Saarbrücken to Frankfurt on Monday?
S: There is a business flight

Can questions in this way be accommodated completely freely? Probably not. It seems that the minimal constraint is that the question should be relevant to the domain, have some relevance to the task at hand. There may also be cases where questions are strongly related. This too is domain-specific; for instance, in the travel agency domain, price and class depend on each other, so that if the user asks about price, the system may well accommodate a question about class and provide information about that.

5.5 Application of the Rules

In this section we look at how the rules are used simultaneously and apply them to some more complex cases.

5.5.1 Simultaneous Rule Application

Consider (52):

(52) S1: Hello, how can I help you?
U1: What is the price of a flight from Paris to London on April fifth?
S2: What class did you have in mind?
U2: I don't know.
S3: BUSINESS class costs ONE THOUSAND euro.
ECONOMY class costs FIVE HUNDRED euro.

In (52:S3) the system enumerates different answers w.r.t. a parameter the user did not specify, here *class*. Given that the QUD is $?\lambda x.price(x)$, the QudTR rule assigns Rheme to the prices, *one thousand euro* and *five hundred euro*, respectively.

How is Focus assigned within the Themes for this example? From a theoretical point of view, there are various ways to tackle this. If the raised but explicitly unanswered question $?\lambda x.class(x)$ results in putting $class(\_)$ into the shared commitments, i.e., it is shared information that the user did not specify constraints on the class, the ComFB rule can assign focus to *business* and *economy*, respectively. If no underspecified question is added to the shared commitments, and the question is simply popped off QUD, domain-knowledge focus assignment could be used.

Another example is the following:

(53) Among shared commitments: *class(business)*, *price(1000,euro)*
U: What does economy class cost?
On QUD: $?\lambda x.(class(economy)\&price(x))$
To generate: *class(economy)&price(200,euro)*
S: ECONOMY class costs TWO HUNDRED Euro

The question on QUD determines $200, euro$ as the Rheme. The ComFB rule establishes 200 and 1000 as alternatives, and *euro* and *euro* as identicals, hence 200 is assigned focus. ComFB also establishes *economy* as an alternative of *business*, and assigns focus to this.

This last example was quite a straightforward example of the rules working the way they were meant to. It should be noted though that the rules have been designed with simple utterances (and turns only consisting of one utterance) in mind. It thus remains an open issue to generalize the rules to more complex sentences and more complex turns.

5.5.2 Multiple Foci

The example just looked at included Foci of different kinds: within the Theme and within the Rheme. Here, we'll take a look at examples involves Foci of the same kind. One example arises in the context of alternative questions and disjunctive statements. Take the following, an example of a helpful alternative question:

(54) On QUD: $?\lambda x.how(x)$
To generate: { ?how(plane), ?how(train) }

Do you want to FLY or do you want to GO BY TRAIN?

Using the representation of alternative questions in the example above, as is the way it's currently done in the GoDiS system, the rules as formulated above can be directly applied also in this situation.

It is also possible to have more than one Focus, of the same kind, within one information unit. Here's an example of two Theme Foci (and one Rheme Focus):

(55) U: How much is a business flight and how much is an economy flight?
S: BUSINESS class and ECONOMY class both cost THREE HUNDRED Euro

Here, the generation of the system utterance is slightly more complex than several of the other examples we've looked at above, since two separate propositions, $class(business)$ & $price(300)$ and $class(economy)\&price(300)$ (and marked for information structure), are turned into a single sentence consisting of, at the syntactic level, a conjunction of noun phrases corresponding to thematic elements. This, of course, is an issue that needs further exploration.

A similar example involving two Rheme Foci of the same kind is the following:

(56) U: What kinds of special meal are served on the flight?
S: The SPECIAL MEALS are VEGETARIAN and GLUTEN-FREE

5.6 Summary

In this chapter, we specified the relation between information structure and context in detail, building on earlier work in the TRINDI project (Engdahl et al., 2000). We have formulated a number of rules for the determination of information structure using the information state. Our rules capture the following ideas: the Theme/Rheme partitioning is derived on the basis of the current question under discussion (the QudTR rule); the Background/Focus partitioning is obtained by comparing the current propositional content with relevant similes, found either in the shared commitments part of the information state (the ComFB rule) or in the representation of the domain (the DomFB rule). We discussed how these rules can be applied to assign information structure is various cases. A experimental implementtion of these rules in GoDiS will be described in Chapter 6.

Chapter 6

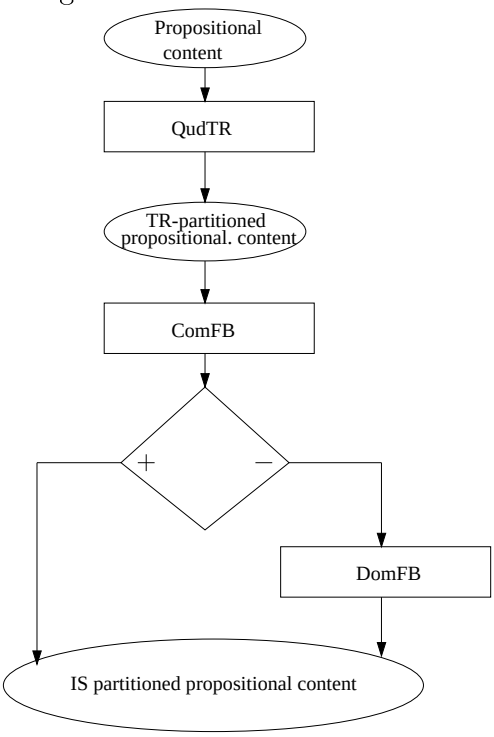
Implementation in GoDiS

We have developed an experimental implementation of the approach to information-structure determination described in Chapter 5. We used the implementation of the GoDiS system developed in the SIRIDUS project as a starting point (Larsson et al., 2002). In this chapter, we describe the details of this implementation. We discuss the implementation of information-structure assignment rules in Sections 6.1, 6.2.1 and 6.2.2. But first, we describe how we fit information structure determination into the process of getting from the propositional content of an utterance to be generated by the system in a praticular information state, to a string of words realizing this content, enriched with annotation that specifies information structure. How the annotation is converted into intonation/prosody specifications for a particular speech synthesis system will be discussed in Chapter 9. In written mode (no speech), the string can of course just be printed as the system's output (without the annotation).

The process of information structure assignment has several phases shown schematically in Figure 6.1. After a move has been selected, its propositional content is sent to a module that assigns information structure partitioning. The module consists of three information assigning rules which are executed one after the other. First, the QudTR rule partitions the semantic form into Rheme and Theme. The, the rule ComFB rule tries to assign Focus to both the Theme and the Rheme of the partitioned form. If it succeeds, the partitioned form is passed to the generation module. If it fails, the rule DomFB rule assigns Focus to the Theme/Rheme partitioned form. The resulting IS-partitioned propositional content is sent to the generation module. We describe the implementation of the individual components in detail in Sections 6.1, 6.2.1 and 6.2.2.

In order to evoke the information structure assignment module for each move which the system plans to produce, we invoke the information-structure assignment in the selection algorithm of GoDiS. This modified selection algorithm is shown in Figure 6.2.

Figure 6.1: Assignment of Information Structure in GoDiS.



The information structure assigned to the propositional content of a dialogue move is encoded by means of operators applied to parts of the propositional content. The operators are: *rh* for Rheme, *foc_rh* for Rheme Focus and *foc_th* for Theme Focus. They are converted into the labels `<RH>, <F_RH> and <F_TH>` respectively.

For instance, an example of a fully partitioned proposition is the one below:

```
class(foc_th(business)), price(rh(foc_rh(1234)))
```

The following utterance labeled with information structure will be generated from it:

```
<F_TH> Business </F_TH> class costs <RH> <F_RH> 1234 </F_RH> </RH> euro.
```

In order to handle the information structure partitioned propositions, the lexicon was changed accordingly by adding new templates. The sections below detail out how we get the IS-partitioning of the propositions.

Figure 6.2: Selection Algorithm including assignment of information structure.

```
selection_algorithm( [
   backupSharedSys,
   if  not ( in( $/private/agenda, A ) and q_raising_action( A ) )
   then ( try select_action )
   else [],
   % select ICM and moves for actions resulting from update
   repeat [( select_icm orelse select_move ),
           try select_tr
           try_select_fb]]  ).
```

6.1 Theme/Rheme Assignment Using QUD (QudTR)

The theme/rheme assignment rule using QUD (QudTR, cf. Section 5.2) takes a proposition and returns a theme/rheme partitioned proposition, in which the Rheme-part is labeled explicitly. For economy reasons, we leave the Theme-part unlabeled.

The QudTR rule is implemented in GoDiS as four disjunctive selection rules 1-4 which fire depending on the semantic form of the move contained in the resource interface variable *content_of_next_moves* and the content of the QUD. Rule 1 is applied if there is a question topmost on QUD which the proposition of the next move in *content_of_next_moves* resolves. If this rule does not apply, another rule can be applied instead. The application of a particular rule here depends on the semantic form of the next move to be generated which is contained in *content_of_next_moves*. Rule 2 is applied if the next move is an answer move representing the results of a database search and resolves a respective question on QUD. Rule 3 is applied if the move in *content_of_next_moves* is a conditional response and resolves a respective question on QUD. Rule 4 provides for the case that the whole proposition should be assigned Rheme. It is applied when the QUD contains no question which the proposition resolves, i.e. if a system utters a proposal or a clarification, or when the system produces a question.

The information structure partitioning assignment itself is carried out by a number of operators which are defined as new operators for GoDiS data types. These operators assign Rheme, Rheme Focus and Theme Focus. For example, the operator *rheme1* shown in Figure 6.3 assigns Rheme to the argument of a proposition. For instance, if the proposition has the form *class(business)*, the operator will assign rheme to *business*, i.e. *class(rh(business))*.

The *rheme1* operator is defined for an answer move in general. Similar operators are defined for the special cases where an answer move conveys the result of a database search, and

Figure 6.3: A Rheme assigning operator

```
operation(rheme1,oqueue([Move|T]), [], oqueue([Move1|T])):-
    Move = answer(A),
    A =..[Functor,Argument],
    A2 =.. [Functor, rh(Argument)],
    Move1 = answer(A2).
```

where the answer is a conditional response (Kruijff-Korbayová et al., 2002b; et al., 2002).

The definitions of the QudTR rules are given below.

In the general case, the resource interfacte variable *content_of_next_moves* contains a move of type *answer* or *inform*. If topmost on QUD is a question which the answer move resolves, the rule applies the operator for assigning Rheme *rheme1* to the argument part of the proposition contained in the variable *content_of_next_moves*.

(RULE 6.1) RULE: **qudTR**
CLASS: select_tr
PRE: { fst($CONTENT_OF_NEXT_MOVES, answer(A))
or fst($CONTENT_OF_NEXT_MOVES, inform(A))
fst($QUD, $?A.B$)
$DOMAIN :: resolves($B$, C)
EFF: { rheme1(CONTENT_OF_NEXT_MOVES)

The second rule is applied to an answer move contained in the *content_of_next_moves* variable. The answer move contains the result of a database search, and QUD contains the question which is resolved by the search result, namely the third argument of the DB search result representation, C. The rule applies the operator for assigning Rheme to the functor of the proposition C.

(RULE 6.2) RULE: **qudTR**
CLASS: select_tr
PRE: { in($CONTENT_OF_NEXT_MOVES, answer(db_entry(A, B, C)))
fst($QUD, $?D.E$)
$DOMAIN :: resolves($E$, C)
EFF: { rheme1(CONTENT_OF_NEXT_MOVES)
pop(QUD)

The third rule specifies the Rheme/Theme assignment for conditional responses. A separate rule is necessary because of the difference in the semantic form of theses answer moves

which is a conditional and consists of two arguments. In this case, the Rheme assigning operator *rheme1* assigns Rheme to the second argument of the conditional response.

(RULE 6.3) RULE: **qudTR**
CLASS: select_tr
PRE: { fst($CONTENT_OF_NEXT_MOVES, answer(implies(A, B)))
fst($QUD, exists($C$))
B=C or B=not(C) }
EFF: { rheme1(CONTENT_OF_NEXT_MOVES)
pop(QUD) }

The last rule provides for the cases where QUD contains no question which a proposition in *content_of_next_moves* can resolve or when the *content_of_next_moves* is an ask move. The rule assigns Rheme to the whole proposition.

(RULE 6.4) RULE: **qudTR**
CLASS: select_tr
PRE: { fst($CONTENT_OF_NEXT_MOVES, answer(A))
or fst($CONTENT_OF_NEXT_MOVES, inform(A))
or fst($CONTENT_OF_NEXT_MOVES, ask(A)) }
EFF: { rheme1(CONTENT_OF_NEXT_MOVES)

An example for the way the rules specifying QudTR work, consider the following dialogue.

(57) U: How much is a cheap flight from London to Paris on April the first?
S: The price is 423 Euro.

In (57S), the system gives an answer which represents the result of a database search resolving a question on QUD $?\lambda x.price(x)$, i.e., rule 2 will apply. The input of the rule is the answer move represented below which is contained in the resource interface variable *content_of_next_moves*.

```
answer(db_entry(set([dept_day(1), month(april), dest_city(paris),
dept_city(london), how(plane), class(economy)]), set([]), price(423)))
```

As a result of applying rule 2, the last argument of the database search result, namely *price(423)*, is passed to the rheme assigning operator *rheme1* which assigns Rheme to its functor. The Theme/Rheme partitioned answer move is sent to the TIS variable *next_moves*:

```
next_moves = oqueue([answer(db_entry(set([dept_day(1),month(april),
dest_city(paris),dept_city(london),how(plane),class(economy)]),set([]),
price(rh(423))))]
```

6.2 Focus/Background Assignment Using Parallellism

The input of the focus/background assignment rules using parallelism is a Theme-partitioned or a Rheme-partitioned information unit/proposition. Their output is a focus/background partitioned proposition, that is, an information unit with labeled Focus-part(s).

6.2.1 Using Shared Commitments (ComFB)

The focus/background assignment rule using shared commitments ComFB is currently applied to two different dialogue moves, namely *inform* and *answer*. The *in_set* operator checks whether the shared commitments contain a proposition that is parallel to the proposition in the inform or answer move. In the definition of ComFBin section 5.3.1, a proposition is parallel to another if the semantic form of the two propositions is such that they have the same functor or the same argument. The current experimental implementation covers only the case where the two propositions have the same functor. For instance, *class(business)* is parallel to *class(economy)*. The operator *focus_arg* assigns Focus to the argument of the functor. In the current experimental implementation, the inform move consists of a list of alternatives. Currently, only one of the alternatives is assigned information structure.

(RULE 6.5)
RULE: **comFB**
CLASS: select_fb
PRE: { fst($CONTENT_OF_NEXT_MOVES, inform(rh([A | _]))) or
fst($CONTENT_OF_NEXT_MOVES, answer(A))
in_set($/SHARED/COM, A) }
EFF: { focus_arg(CONTENT_OF_NEXT_MOVES) }

To give an example for the application of the ComFB rule, consider the dialogue in (58).

(58) U: A cheap flight from London to Hongkong
S: What month do you want to leave?
U: Can I fly in april?

S: Not if you want to travel economy class.
S: We can offer business class. Are you interested?

The last move in (58) is an inform move conveying the proposition *class(business)*:

```
inform(class(business))
```

It is already Theme/Rheme partitioned by one of the QudTR rules:

```
inform(class(rh(business)))
```

This Theme/Rheme partitioned inform move is the current content of the TIS variable *content_of_next_moves* and as such the input of the ComFB rule. ComFB is applies, since SHARED/COM contains the proposition *class(economy)* which is parallel to the proposition *class(business)*. The focus assignment operator *focus_arg* assigns Focus to the argument of the proposition *inform(class(rh(business)))*, namely the Rheme partitioned part of the propostition. The so partitioned proposition is send to the TIS variable *next_moves*:

```
next_moves = oqueue([inform([rh([class(foc_rh(business))|A])])])
```

6.2.2 Using Domain Knowledge (DomFB)

The focus/background assignment rule using domain knowledge DomFB is implemented in GoDiS by specifying three separate selection rules. The general case is covered by rule 6 which takes the proposition of the move contained in the *content_of_next_moves* which is an *answer* or *inform* move and tests whether it is a proposition of the domain, that is, whether the proposition has an alternative in the domain. As a result of the rule, the operator for focus assignment *focus_arg* assigns Focus to the argument of the already Theme/Rheme partitioned proposition.

(RULE 6.6) RULE: **domFB**
CLASS: select_fb
PRE: $\begin{cases} \text{fst}(\$\text{CONTENT\_OF\_NEXT\_MOVES}, \text{answer}(A)) \text{ or} \\ \text{fst}(\$\text{CONTENT\_OF\_NEXT\_MOVES}, \text{inform}([A])) \\ \$\text{DOMAIN} :: \text{proposition}(A) \end{cases}$
EFF: $\{$ focus_arg(CONTENT_OF_NEXT_MOVES)

The more specific rule 7 covers the case where the answer move is a non-conditional answer. As a result of applying the rule, the operator *focus_arg* assigns focus to the argument of the already Theme/Rheme partitioned proposition.

(RULE 6.7) RULE: **domFB**
CLASS: select_fb
PRE: { in($CONTENT_OF_NEXT_MOVES, answer(db_entry(A, B, C)))
$DOMAIN :: proposition($C$) }
EFF: { focus_arg(CONTENT_OF_NEXT_MOVES)

Another specific rule is provided for conditional responses. The rule applies two focus assigning operators, *focus_th_arg* and *focus_rh_1*, to the two different arguments of the already Theme/Rheme partitioned proposition of the answer move. The operator *focus_th_arg* assigns Focus to the Theme part of the proposition, and the operator *focus_rh_1* assigns Focus to the Rheme part of the proposition.

(RULE 6.8) RULE: **domFB**
CLASS: select_fb
PRE: { fst($CONTENT_OF_NEXT_MOVES, answer(implies(A, B)))
$DOMAIN :: proposition($A$)
$DOMAIN :: proposition($B$) }
EFF: { focus_th_arg(CONTENT_OF_NEXT_MOVES)
focus_rh_1(CONTENT_OF_NEXT_MOVES)

In the current experimental implementation, the DomFB-rules assign the same information structure partitioning as the ComFB rule even if there is no relevant information contained in the shared commitments. A topic for future work is to make the rule more specific by constraining the alternatives that a proposition can have in the domain.

Another topic for future work is to extend the DomFB rules to cover also ask moves. In the case of ask moves, the QudTR rule 4 assigns Rheme to the whole question. The DomFB rule will then assign Rheme Focus to the element of the proposition which corresponds to what the question is about, i.e. the functor. For instance, a question like *Which city do you want to go to?* represented as $?Xdestination\_city(X)$ will be assigned Rheme by QudTR rule 4 (represented as $rh(?Xdestination\_city(X))$) and Rheme Focus by DomFB (represented as $rh(?X(foc\_rh(destination\_city))(X))$).

The last example shows the assignment of both Theme Focus and Rheme Focus. Consider the following dialogue.

(59) S1: Hello, how can I help you?
U1: What is the price of a flight from Paris to London on April fifth?
S2: What class did you have in mind?
U2: I don't know.
S3: BUSINESS class costs ONE THOUSAND EURO.
ECONOMY class costs FIVE HUNDRED EURO.

The first utterance in (59S3) is an answer move conveying the propositions *class(business)* and *price(1000)*. It is already Theme/Rheme partitioned by one of the QudTR rules:

```
answer(db_entry(set([dept_day(1),month(april),
dest_city(paris), dept_city(london),how(plane)]),
set([class(business)]),
price(rh(1234))))
```

This Theme/Rheme partitioned answer move is the current content of the TIS variable *content_of_next_moves* and as such the input of the Focus/Background assigning rule. In this case, DomFB is applied since SHARED/COM contains no proposition *class(economy)* which is parallel to the proposition in the answer *class(business)*. Since the answer is non-conditional, rule 7. is applied. The focus assignment operator *focus_arg* assigns Focus to the argument of the proposition *price(rh(1234))*, which is the Rheme part of the propostition. The resulting partitioned proposition is then send to the TIS variable *next_moves*:

```
next_moves = oqueue([answer(db_entry(set([dept_day(1),month(april),
dest_city(paris), dept_city(london),how(plane)]),

set([class(foc_th(business))]),

price(rh(foc_rh(1234))))])
```

6.3 Summary

In Chapter 5 we specified a set of rules determining how an information structure partitioning can be derived from the information state. In this chapter, we presented an experimental implementation on these ideas, developed in GoDiS. The rules assigning information structure are implemented as a module that takes as input the propositional

content of a dialogue move, and returns this content partitioned according to the current question under discussion (the QudTR rule, and the contents of the shared commitments (the ComFB rule) and the domain knowledge (the DomFB rule). The partitioned content that this module outputs serves as input to the generation of the surface realization, which produces a string of words along with an annotation of the information structure partitioning. This information structure annotation uses an internal set of labels. These are subsequently converted into markup suitable for text to speech synthesis. The generation of contexually varied spoken output in GoDiS using two off-the-shelf text-to-speech synthesis systems is described in Chapter 9.

Chapter 7

State of the Art in Speech Synthesis

Speech synthesis can be defined as the automatic generation of speech by artificial means. The ultimate goal in this art is the emulation of human speech at the highest level possible of intelligibility and naturalness. Advances in this field have provided systems with a high degree of intelligibility; however voice quality and naturalness are still far from ideal.

Different strategies and techniques have been implemented throughout the last few decades. Some have rendered better results than the rest in some respects, but in general, there is always a trade off between voice quality, computational complexity, intelligibility, coverage, etc.

Speech Synthesis Systems can be generically divided into two groups:

- Rule–based synthesis, and
- Data–driven synthesis.

In the former type, there is a set of manually defined rules that drive the system. In the latter type, patterns and parameter values are drawn from data (i.e. corpora) and define the systems behavior.

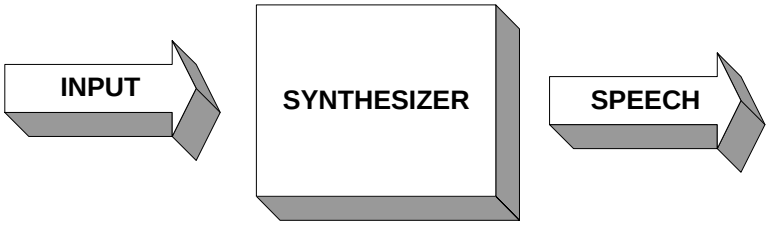
According to the quality of their output and their flexibility, systems can be classified into four main families (Huang et al. 2001):

1. Limited–domain waveform concatenation
2. Concatenative synthesis with no waveform modification

3. Concatenative synthesis with waveform modification

4. Rule–based systems

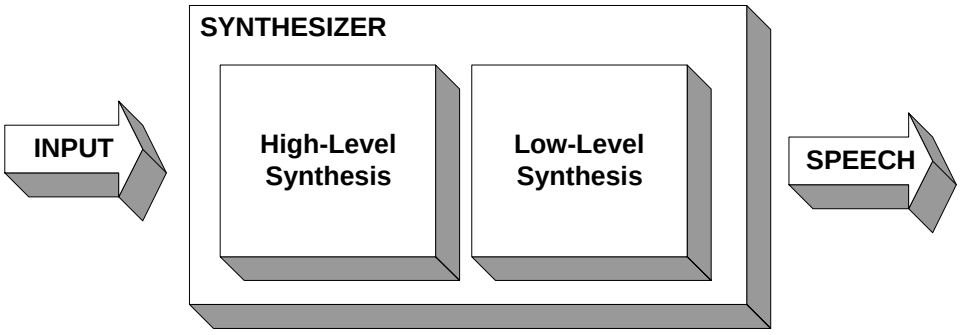
In general, the process of synthesizing speech can be outlined in very simple blocks:



In a Spoken Dialogue System, the input provided to the synthesizer can be "canned", i.e., there are written sentences that are directly sent to the synthesizer in a given case, or they can also be automatically generated, that is, the Dialogue System incorporates a Natural Language Generator that, given certain instructions, will generate the input for the synthesizer.

Both types of input can also incorporate tags that may help the synthesizer generate more appropriate utterances. These tags usually conform to one of the speech synthesis mark-up standards that will be discussed in Section 7.3.

There are two main phases in the synthesis process, which are normally called high and low level synthesis:



During the high–level phase, the input text is transcribed into some linguistic representation. This phase can also be divided into three main phases: text pre–processing, pronunciation generation and prosody generation. We describe these phases in more detail in Section 7.2

It is within the low–level phase that the actual waveforms are generated. The linguistic representation generated during the high–level will be translated into a waveform. This

step is critical in the synthesis process and several strategies have been implemented. These approaches will be discussed in Section 7.1.

Both phases present important challenges that have a direct impact on the quality of synthetic speech. We describe the low–level first, and then the high–level.

7.1 Low–Level Synthesis

There are three main groups:

7.1.1 Articulatory Synthesis

These systems model the human speech production system directly (O.Saughnessy 1987, Witten 1982, Donovan 1996), that is, the human articulators and vocal cords. It is a rule–based synthesis approach that often use a total of only 15 parameters to drive a formant synthesizer (Huang et al 2001).

There are quite a few challenges to overcome in this approach: data collection, rule generation, accurate modeling of the tongue, vocal cords and resonator, etc.

Although this method is very promising, its level of complexity is very high, which translates into a computational load increase with respect to other methods, and a lower degree of development. The state of the art in articulatory synthesis is not comparable to that of formant or concatenative systems.

7.1.2 Formant Synthesis

It is a very widespread method based on a source–filter model of speech. It can generate an infinite number of sounds, which implies greater flexibility in the synthesis process. Most rule–based synthesizers use formant–based synthesis.

Intelligible speech requires a minimum of three formants, whereas high quality speech requires five. A set of rules is used to determine which parameters are required (Allen et al. 1987). Some of the input parameters may be (Holmes et al 1990):

1. Voicing fundamental frequency (F0)

2. Voiced excitation open quotient (OQ)
3. Degree of voicing in excitation (VO)
4. Formant frequencies and amplitudes (F1...F3 and A1...A3)
5. Frequency of an additional low-frequency resonator (FN)
6. Intensity of low- and high-frequency region (ALF, AHF)

There are two main methods that may also be combined into a hybrid one. The parallel model is normally used for the synthesis of nasals, fricatives and stops and the cascade model for all voiced sounds (Huang et al. 2001)

Cascade Method

It is a series of resonators connected to each other where the output of each is the input of the next. This method only needs formant frequencies as control information. Its main advantage is that formant amplitudes for vowels do not need individual control (Allen et al. 1987).

Parallel Method

In this model, resonators are connected in parallel. The signal is sent to all resonators at once, rather than to the first one in the series (cascade system). Their outputs are then added taking adjacencies into account.

Hybrid Method

This is a combination of the two techniques above mentioned. It takes advantages of the best qualities of both methods and joins them to produce superior quality synthetic speech.

Data–Driven Formant Generation

Although most models are based on parameter values generated by rules, there are also data–driven methods. Formant generation is based on a HMM that emits three formant frequencies and their bandwidths to a cascade formant synthesizer (Huang et al. 2001).

7.1.3 Concatenative Synthesis

As its name may indicate, Concatenative Synthesis is the *"concatenation"* of pre–recorded speech segments. Since the prime material used is actual human speech, the outcome is expected to be natural and does not require manual tuning or rules at segment level.

There are nonetheless problems that arise from concatenating speech fragments that were not adjacent to each other when they were produced. This problem is known as *"coarticulation"* and may originate spectral or prosodic discontinuities that make the utterance sound unnatural.

7.2 High–Level Synthesis

This phase can also be called "*text-to-phonetic*" or "*grapheme–to–phoneme*" conversion. Although some of the challenges in this phase are common to all languages, some problems are language–specific. For instance, phonetic languages, such as Spanish or Italian, present fewer problems in the pronunciation generation phase than non–phonetic languages such as English or French.

There are three main phases: text pre-processing, pronunciation generation and prosody generation. We discuss them below in more detail.

7.2.1 Text Pre–processing

In this phase the text is analyzed and all ambiguous features are pondered and translated to unambiguous strings. Some of these ambiguities are language specific.

Numerals, abbreviations and special characters are classic problems. For instance in English, numerals will be read differently depending on what they mean:

- Christopher Columbus discovered America in **1492**. *(fourteen ninety-two)*
- There are **1492** seats on section C. *(one-thousand four-hundred and ninety-two)*

Abbreviations can expand into different words depending on the context:

- I live on 2465 Churchill **Dr.** *(drive)*
- **Dr.** Livingston, I suppose. *(doctor)*

Acronyms may be spelled, read as a common word or expanded into what they stand for:

- **TTS** Synthesis *(T-T-S or Text to Speech)*
- **JPEG** *(J-peg)*
- **AIDS** is a relatively new disease *(aids)*

7.2.2 Pronunciation generation

The main challenges in this phase are common to many languages.

The pronunciation of words in context often differs from the pronunciation of the same words in isolation. This is normally due to the phonetic context generated in the utterances. Clear examples are:

- Articles followed by words beginning with open vowels:
  - **The** argument
  - **The** cat
  - **The** internet
- Inflectional suffixes in voiced vs. voiceless contexts
  - Bug**s** *(voiced)*
  - Butt**s** *(voiceless)*
- Special character combinations
  - Thre**sh**old *(shh)*
  - Ca**sh**ew *(sh)*
- Proper Names
  - Nice
  - Nice

- Homographs
  - They will **lead** us to a new and better place *(to lead)*
  - The bag seems to be full of **lead** *(metal)*

7.2.3 Prosody Generation

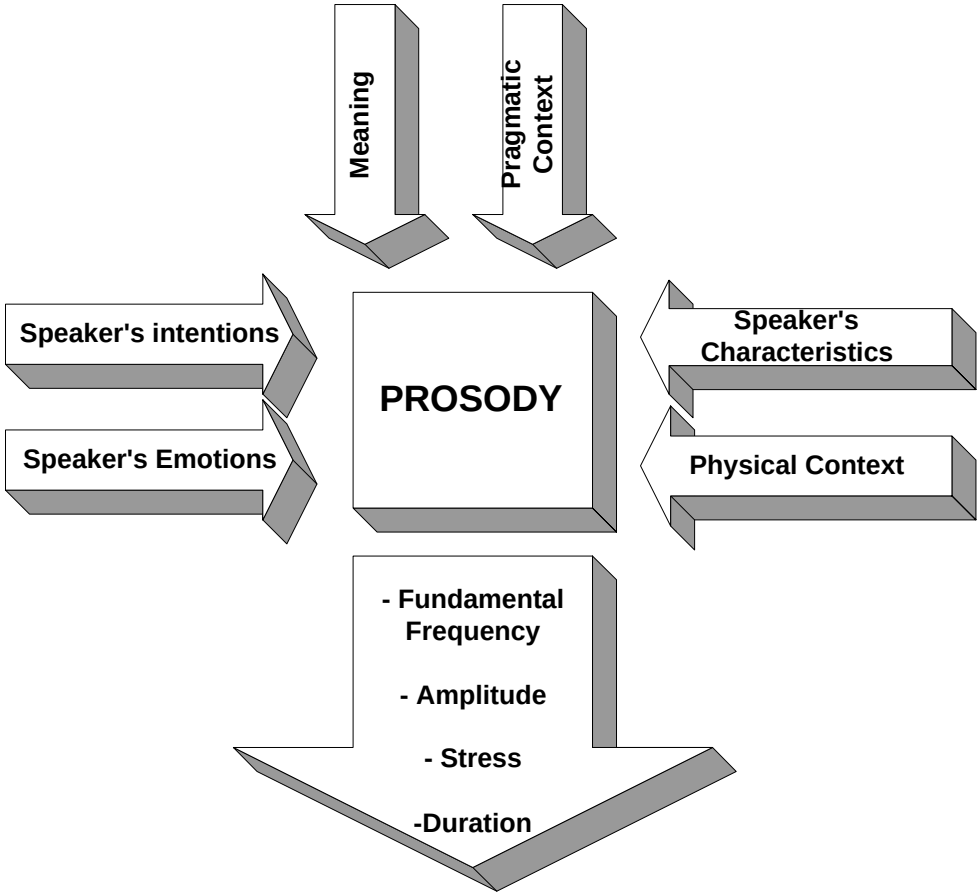
Generating the correct prosody from written text is one of the most challenging problems in synthetic speech.

The intonation of an utterance can be thought of as the representation of the fundamental frequency changes throughout the utterance, and it is a combination of many factors. Prosody has specific functions in speech communication, and one of its most relevant ones is "focus". Among other aspects, prosody provides information on focus, word relationships, contrast, end of speech, sentence segmentation, etc.

The prosodic contour of continuous speech may depend on many aspects such as speaker's intentions, attitude, characteristics and emotions, meaning of the utterance and the pragmatic context of the conversation. Although written text *per se* does not contain much information about all these aspects, it is possible to provide the synthesizer with at least some of this information in order to generate more natural and appropriate prosodic contours.

Parameters

- Pauses
- Pitch
- Rate/Duration
- Volume
- Melody
- Tones



Emotions

According to several authors (Abadjieva et al. 1993, Murray et al. 1993), the voice parameters influenced by emotions can be classified in three types:

- *Voice quality* which contains largely constant voice characteristics over the spoken utterance, such as loudness and breathiness. For example, angry voice is breathy, loud, and has a tense articulation with abrupt changes while sad voice is very quiet with a decreased articulation precision.
- *Pitch contour* and its dynamic changes carry important emotional information, both in the general form for the whole sentence and in small fluctuations at word and phonemic levels. The most important pitch features are the general level, the dynamic range, changes in overall shape, content words, stressed phonemes, emphatic stress, and clause boundaries.

- *Time characteristics* contain the general rhythm, speech rate, the lengthening and shortening of the stressed syllables, the length of content words, and the duration and placing of pauses.

According to the same authors, there are five primary emotions:

- anger
- fear
- happiness
- sadness
- disgust

And the secondary states are:

- whispering
- shouting
- grief
- tiredness

Each of these states have particular characteristics that can be emulated (speed, loudness, dynamic vs. static changes, etc)

Information–based Prosody

In most current systems, intonation is kept "acceptable neutral".

> "Acceptable intonation must be plausible, but need not be the most appropriate intonation for a particular utterance: no assumption of understanding or generation by the machine need be made. Neutral intonation does not express unusual emphasis, contrastive stress or stylistic effects: it is the default intonation which might be used for an utterance out of context. (...) This approach removes the necessity for reference to context or world knowledge while retaining ambitious linguistic goals." (Monaghan 1989)

Although this is a reasonable approach to handle situations that do not require non-neutral intonation, there is a great number of situations that would normally imply a particular intonation in human-human discourse.

To recreate these differences and distinguish these cases artificially, the use of information at different levels is essential:

- domain
- general purpose of the dialogue
- dialogue stage
- dialogue history
- register

in addition to the information related to the particular words and structure chosen:

- syntactic construction
- stress pattern
- number of words
- focus
- etc.

All this information is available to the system and can be used to generate more appropriate prosodic patterns. In constraint domain systems, a certain set of prosodic possibilities can be foreseen. The general purpose of the dialogue (to facilitate the purchase of a certain product, to provide or collect information, etc.) may also help determine a certain set of prosodic patterns. All the factors mentioned above may be taken into account to generate more natural prosodic patterns. Text analysis is one of the major obstacles to achieve natural synthetic speech; in this case, the text structure is pre-generated and therefore no structural ambiguity exists.

7.3 Speech Mark–up Languages

In terms of the mark–up languages, although there are several candidates, we will focus on two: SABLE and SSML.

7.3.1 SABLE

It is a consortium aimed at providing a single standard for speech synthesis markup, which evolved as an initiative to combine three existing speech synthesis markup languages: SSML, STML and JSML. Its features are:

- Emphasis
- Pause (prosodic breaks)
- Pitch
- Rate
- Volume
- Prerecorded material
- Pronunciation hints
- Language
- Gender/age/voice font
- Non–standard extensions

7.3.2 SSML

The first attempt to develop a TTS markup language was called SSML (Speech Synthesis Markup Language), which describes mark–up for generating synthetic speech via a speech synthesiser, and forms part of the proposals for the W3C Speech Interface Framework

Although SSML seems to be more complete and powerful than SABLE, the latter has been implemented in Festival 1.4.1, which is available for research and allows for the kind of manipulations above mentioned. Given that SSML may prove to be a more convenient standard (even though it may need to be extended for the purposes above mentioned), it may be more convenient to our own SSML/SABLE based markup that can be translated to whatever standard may be necessary.

Chapter 8

Speech Synthesis Systems

Many synthesizers are nowadays available, some of them also for research purposes. In (Hieronymus et al., 2002), several text to speech synthesis systems were compared from the viewpoint of their usability in the SIRIDUS project. For experimenting with contextually determined varied speech output we have narrowed the selection down to three systems, namely FESTIVAL (Section 8.1), MARY (Section 8.2) and Telefónica's TTS (Section 8.3). This choice was motivated by the possibility to control intonation in FESTIVAL and MARY by using higher level annotation. Naturally, Telefónica uses their own TTS system for commercial purposes.

8.1 The FESTIVAL TTS

Festival is a general multi–lingual speech synthesis system developed at CSTR. It offers a full text to speech system with various APIs, as well an environment for development and research of speech synthesis techniques. It is written in C++ with a Scheme–based command interpreter for general control[1].

8.1.1 Features

- English (British and American), Spanish and Welsh text to speech
- Externally configurable language independent modules:

[1] The systems description as well as the enumeration of features has been copied from its original website http://www.cstr.ed.ac.uk/projects/festival/

  - phonesets
  - lexicons
  - letter–to–sound rules
  - tokenizing
  - part of speech tagging
  - intobeginnation and duration
- Waveform synthesizers:
  - diphone based: residual excited LPC (and PSOLA not for distribution)
  - MBROLA database support (http://tcts.fpms.ac.be/synthesis/mbrola.html)
  - distributed under a free X11–type licence
  - generalization of stats modules, ngram, CART, wfst with viterbi so they can be shard more easily
  - Initial JSAPI support
  - XML load for Relations
- Portable (Unix) distribution
- On–line documentation
- SABLE markup (http://www.cstr.ed.ac.uk/projects/sable/), Emacs, client/server (including Java), scripting interfaces.

Festival's basic sound unit is the phoneme. It can support several different phone sets which can be mapped from one to another. Phones can have several phonological features associated.

Its lexicon contains word pronunciations and a customisable list of words that are particular to a user or application (addenda).

Festival incorporates a discourse tagger, SOLE. This module adds information that will be useful to generate the appropriate prosodic contours. The text may however be already tagged, i.e., labelled according to the XML based formalism called SABLE. If the text is untagged, sentence divisions are assigned by an algorithm whose heuristics are based on punctuation, capitalization and white space.

Festival performs its linguistic analysis by means of a n–gram based POS tagger, which also resolves most cases of homograph disambiguation. It also incorporates Letter–To–Sound rules to generate the pronunciations of words absent from the lexicon. This LTS rules may be stochastically generated or written by hand.

8.1.2 ToBi Intonation Annotation in FESTIVAL

An experimental set of patches for FESTIVAL developed by Robert Clark at the University of Edinburgh[2] allows to annotate FESTIVAL input with various higher levels of information, including speech-act type and turn-taking information, but also the annotation of the Theme/Rheme partioning according to Steedman (Steedman, 2000a) and of additional intonation information coded in terms of a ToBI-based markup.

8.2 The MARY TTS

MARY[3] is a complete Text-to-Speech Synthesis System. Currently, two versions are available: for German and for English.[4] MARY is developed as a collaborative project of DFKI's language technology lab and the Institute of Phonetics at Saarland University. MARY is designed to be highly modular, with a special focus on transparency and accessibility of intermediate processing steps. MARY allows step-by-step processing with an access to partial processing results, which it does not only display, but also allows the user to modify, whereby the user can interactively explore each processing step. This makes MARY a suitable tool for research and development.

Moreover, MARY allows the user to specify partial annotation at any level (for example, the user can specify just the intonation annotation, and it is even possible to only specify that partially), and the rest is then calculated automatically. This is a feature we exploit when using MARY to generate spoken output in GoDiS.

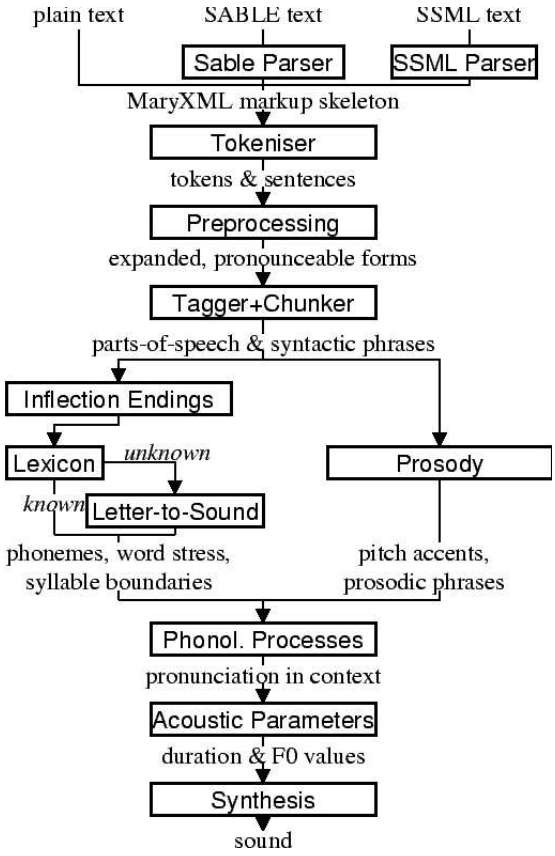
From the viewpoint of natural language processing, the TTS processing architecture of MARY (cf. Figure 8.1) is similar to that of FESTIVAL. It has the following four parts: the preprocessing or text normalization; the natural language processing, doing linguistic analysis and annotation; the calculation of acoustic parameters, which translates the linguistically annotated symbolic structure into a table containing only physically relevant parameters; the synthesis, transforming the parameter table into an audio file. We describe the modules briefly below.

[2] *http://www.cstr.ed.ac.uk/~robert/*

[3] This section has been adapted from the overview of the MARY system available through the website at *http://mary.dfki.de*, and the more detailed description of the MARY system and its use for research, teaching and development in (Schröder and Trouvain, 2001).

[4] The English version of MARY uses FreeTTS to produce the synthesized speech. The full flexibility of different levels of processing is not yet supported in the current English version.

Figure 8.1: The architecture of the MARY TTS system for German



The preprocessing The preprocessing or text normalization includes the tokenizer, abbreviation expansion, and numeral expansion. At the same time, a rudimentary internal XML structure is built around the input text, eventually translating any SABLE annotation that may be given in the input text.

The natural language processing The natural language processing is responsible of the calculation of speech-relevant data out of the written input text, viz. phone symbols and intonation labels. In a first NLP step, part of speech labeling and shallow parsing (chunking) is performed. Then, a lexicon lookup is performed in the pronunciation lexicon; unknown tokens are morphologically decomposed and phonemized by grapheme to phoneme (letter to sound) rules. Independently from the lexicon lookup, symbols for the intonation and phrase structure are assigned by rule, using punctuation, part of speech info, and the local syntactic info provided by the chunker. Finally, postlexical phonological rules are applied, modifying the phone symbols and/or the intonation symbols as a function of their context.

The NLP analysis is organized in a modular way, containing the following components:

- part of speech tagger
- chunker (a partial syntactic analysis)
- grapheme to phoneme conversion using
  - a lexicon for the known tokens
  - grapheme to phoneme rules for the unknown tokens, using a morphological analysis
  - syllabification, word stress and phonologic rules
- intonation annotation using ToBI (in particular, GToBI for German (Grice et al., to appear); the English version does not yet support ToBI annotation)
- postlexical phonological rules

The output of the NLP component is a rich MaryXML structure, the structure of which is defined in the MaryXML data type definition. An example is shown in Figure 8.2.

The component that is of crucial relevance for our current work is that of intonation annotation. MARY uses an adaptation of ToBI[5] for German (Grice et al., to appear). MARY supports the full GToBI inventory of tones:

Pitch accent tones: H*, !H*, ˆH*, L*, L+H*, L*+H, L+!H*, L*+!H, L+ˆH*, L*+ˆH, H+L*, H+!H*, H+ˆH*, !H+!H*, ˆH+!H*, ˆH+ˆH*

Boundary tones: H-, !H-, H-%, !H-%, H-ˆH%, !H-ˆH%, L-H%, L-%

Break indices in MARY are used as follows: "2" is a potential boundary location (which might be "stepped up" and thus realized by some phonological process later on); "3" denotes an intermediate phrase break; "5" and "6" (not part of GToBI) represent sentence-final and paragraph-final boundaries.

The prosody module assigns the symbolic GToBI labels. In a later step, these are translated into concrete F0 targets and pause durations. The prosody rule used in MARY were derived through corpus analysis and are mostly based on part-of-speech and punctuation

[5]For more information about ToBI ("Tones, Breaks and Indices") cf. *http://www.ling.ohio-state.edu/~tobi/.*

Figure 8.2: Example of text annotation in the MaryXML format.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE maryxml SYSTEM "http://mary.dfki.de/lib/MaryXML.dtd">
<maryxml>
<div>
<phrase>
<t g2p_method="lexicon" pos="ART"
       sampa="'aI-n@" syn_attach="1" syn_phrase="NP">
Eine
</t>
<t accent="l+h*" g2p_method="lex_adj" pos="ADJA"
   sampa="'?{C-t@" syn_attach="0" syn_phrase="NP">
echte
</t>
<t accent="h+l*" g2p_method="lexicon" pos="NN"
   sampa="h{-'raUs-fO6-d6-rUN" syn_attach="0" syn_phrase="NP">
Herausforderung
</t>
<t pos="$." syn_attach="2" syn_phrase="_">
.
</t>
<boundary breakindex="6" tone="l-l%"/>
</phrase>
</div>
</maryxml>
```

information. Some part-of-speech, such as nouns and adjectives always receive an accent; the other part-of-speech are ranked hierarchically (roughly: full verbs ¿ modal verbs ¿ adverbs), according to their aptitude to receive an accent. This ranking comes into play when the obligatory assignment rules do not place any accent inside some intermediate phrase. After determining the location of prosodic boundaries and pitch accents, the actual tones are assigned according to sentence type (declarative, interrogative-W, interrogative-Yes-No and exclamative). For each sentence type, pitch accent tones and intonation phrase boundary tones are assigned.

It is the placement and assignment of pitch accent tones and boundary tones that we modify in accordance to the information state, when we produce contextually varied speech in GoDiS using the MARY system (see Section 9.1.1).

The Calculation of Acoustic Parameters The richly annotated input is then translated into an acoustic parameter file, by applying a model for duration (the so-called Klatt Rules adapted for German) and for intonation (a ToBI based approach, translating intonation symbols into targets on declination lines that can be attributed precise frequency values).

The output is a parameter file as used in one way or another by many speech synthesis systems. For the time being, MARY uses MBROLA as a synthesis system, so the parameter output format is the MBROLA input format. Every phone symbol is assigned a duration in milliseconds; some phone symbols are assigned a (time,frequency) target, where time is in percent of the phone duration and frequency is in Hertz. As for MBROLA, a male and a female voice exists.

The synthesizer Finally, the synthesizer creates a sound file from a phoneme string. At present, we use MBROLA; but other synthesis systems should be quite easy to "plug in". Several audio formats can be generated, including 16 bit wav, aiff, and au, with 16000 Hz sampling rate.

8.3 Telefónica's TTS for Spanish

Telefónica's text-to-speech system is a speech synthesizer based on unit concatenation available for different languages and platforms. The possible languages are Spanish (Castilian, South American and Peruvian), Catalan, Galician, Basque and Brazilian Portuguese. It is integrated in Telefónica's intelligent network platform, with Unix (Linux and Unixware) and Windows (NT) versions. There is also a version for SAPI 5.0 (Microsoft Speech SDK), and a development version for Solaris.

The system is based on unit concatenation. It makes use of a combination of diphones, triphones and even bigger units to improve synthesized voice quality. There is a LPC-based version which is used when important memory restrictions are imposed, and a sinusoide-based version with a higher quality.

The Castilian Spanish version provides several speakers based on human voices more or less elaborated prosodically. It is possible to change the speaker on the fly during dialogue and the developer could therefore choose a male voice for some part of the dialogue and a female voice for another part. This is useful to make it easier for the user to distinguish between different information. For example, in an e-mail reading system, the dialogue could be held with one voice and the e-mail could be read with another voice or in another language if that is the case.

The TTS system provides tables for the treatment of abbreviations, acronyms, foreign words, numbers, dates, hours, temperatures etc. The developer has the possibility to alter these tables and, for example, add the correct pronunciation of an English word, e.g. transcribe Shakespeare.

The system also provides a set of labels which allow the user to control certain TTS production parameters, as we will see in Section 9.2. These are non standardized labels, and are sent to the TTS directly included in the text to be synthesized. This means that, if a system makes use of another markup system (for instance, JSML or SABLE) a markup interpreter should be used.

Another interesting feature of this TTS system is that it can notify the user (i.e. the dialogue manager) when particular parts of the text have been synthesized. This makes possible to synchronize speech output with other events. For example, this could be used in a multimedia application showing pictures corresponding to the words or text parts the system is reading.

Chapter 9

Producing Varied Synthesized Speech Output

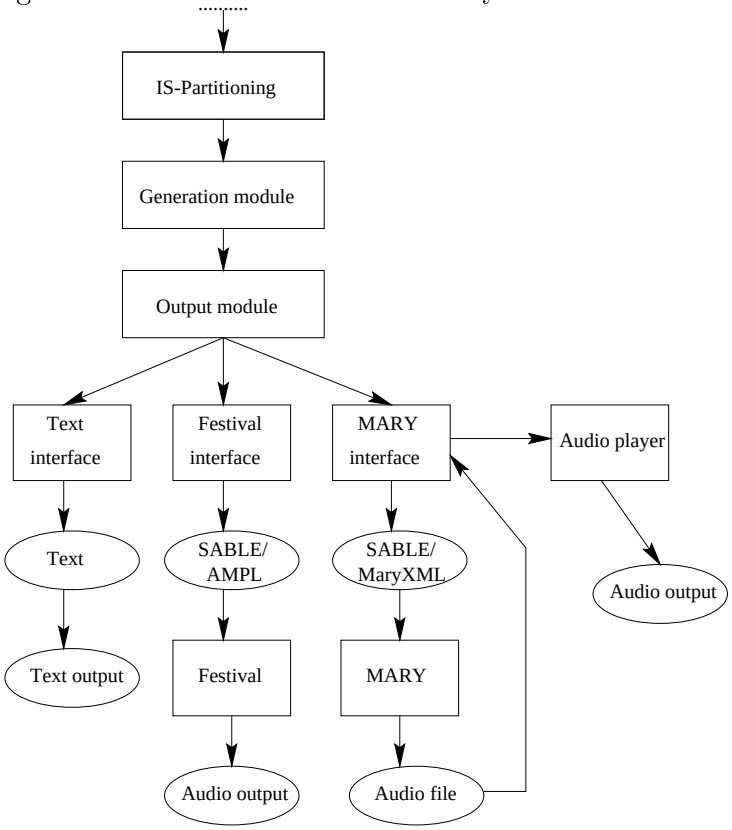
In this chapter we describe how contextually varied speech output is produced using off-the-shelf speech synthesis systems in various versions of the dialogue systems developed in the SIRIDUS project. In Section 9.1 we present varied intonation production in GoDiS in German and English, using the MARY TTS system (Section 9.1.1) and the FESTIVAL TTS system (Section 9.1.2). These implementations realize the spoken output based on the information structure partitioning assignment described in Chapters 5 and 6. In Section 9.2 we describe how different possibilities of the Telefónica's TTS system have been used to improve the system output. In Section 9.3 we present the design of a set of strategies for experimenting with and evaluating varied synthesized speech output in the Delfos 2 system using FESTIVAL TTS.

9.1 Varied Synthesized Speech Output in GoDiS

We integrated both the MARY TTS and FESTIVAL systems into GoDiS, and defined various mappings from our internal information structure annotation to intonation annotation formats used by these systems. This enables us to experiment with the following different versions:

- MARY for German with either SABLE or GToBI intonation annotation
- MARY for English with SABLE intonation annotation
- FESTIVAL for English with either SABLE or the AMPL annotation (including ToBI)

Figure 9.1: Interface between TTS systems and GoDiS.



The possibilities of output from GoDiS we experiment with, using interfaces with various TTS systems is shown schematically in Figure 9.1. They are briefly described below.

The MARY system supports plain text, the MaryXML format or the SABLE format as input. The interface between GoDiS and MARY works as follows: When MARY is used with MaryXML input (e.g., for German), the output module of GoDiS takes a string annotated with information structure partitioning and calls the interface between MARY and the MaryXML markup language. This interface takes the GoDiS-internal information structure partitioning tags and converts them into tags of the GToBI markup in MaryXML. The result of the conversion is saved into a MaryXML file. When MARY is used with SABLE input (e.g., for English), the difference is that the interface between MARY and SABLE takes the GoDiS-internal information structure partitioning tags and converts them into tags of SABLE, and the result of the conversion is saved into a SABLE file. Then, the output module calls a Unix/Linux shell. A demo socket developed at the DFKI sends the MaryXML/SABLE file to a MARY server. The server converts the MaryXML/SABLE file into an audio file and sends it back to the output module. The output module calls a Unix/Linux shell again and plays the audio file using an audio application

The interface between GoDiS and FESTIVAL works as follows: The output module of GoDiS takes a string annotated with information structure partitioning and calls the interface to the SABLE markup language or the ToBI-based AMPL markup language. This interface takes the GoDiS-internal information structure partitioning tags and converts them into the corresponding SABLE/AMPL tags. The result of the conversion is saved into a SABLE/AMPL file. Unlike MARY which runs remotely on the Mary server, FESTIVAL is run locally. So the output module of GoDiS calls a Unix/Linux shell and sends the SABLE/AMPL file to FESTIVAL, which synthesizes and outputs the synthesized speech. FESTIVAL could also be used as a server, in which case it would send an audio file with the synthesized output back to GoDiS.

In the following two sections, we illustrate the conversions to the ToBI intonation annotation in the MaryXML format employed in the MARY TTS system for German MaryXML format, and the conversions to the SABLE annotation employed in both MARY for English and in FESTIVAL. The output audio files can be accessed at the project website *http://www.coli-uni-sb.de/cl/projects/siridus.html*.

9.1.1 Varied Synthesized Speech Output in GoDiS Using ToBI

In this section we illustrate the use of the ToBI intonation annotation to control the intonation of synthesized spoken output of GoDiS, on the basis of the information structure partitioning described in Chapter 6. We illustrate the conversions from the GoDiS internal information structure annotation to ToBI on the MARY TTS system for German developed at the DFKI (German Research Center for Artificial Intelligence) and the Phonetics Department of the University of Saarland (cf. Section 8.2 for details).[1]

As we have seen in Chapter 6, the output of the GoDiS generation module is a string annotated with information structure partitioning. For instance, the information structure partitioning of (60:S1) assigned by the theme/rheme assignment rule QudTR and the focus assignment rule ComFB will be represented as the tagged string in (61).

(60) U1: Wieviel kostet die zweite Klasse? (How much is the economy class?)
S1: Der Preis ist DREIHUNDERT Euro. (The price is three hundred Euro.

[1]We only obtained the experimental set of patches developed by Robert Clark at the University of Edinburgh recently, and have not therefore been able to experiment much with them. Thee patches referred to as "AMPL" allow to annotate FESTIVAL input directly with the Theme/Rheme partitioning according to Steedman (Steedman, 2000a) and with additional intonation information coded in terms of a ToBI-based markup. The conversions of our internal IS-partitioning annotation into the AMPL annotation format for English are similar to those we present for MaryXML. Besides the GToBI markup that MaryXML supports, AMPL also supports other higher level information such as speech-act type and turn-taking information.

(61) `Der Preis ist <RH> <F_RH> dreihundert </F_RH> </RH> Euro.`

The string with the GoDiS internal information structure partitioning representation is sent to the GoDiS output module and from there to the MARY interface. This interface converts the GoDiS internal information structure partitioning tags into GToBI tags in MaryXML tags which are then stored in a MaryXML file. The GToBI converted representation for (61) in MaryXML is (62). As already pointed out, in GToBI, Rheme-focus is tagged with H* and Theme-focus is tagged with L+H*. Additionally, if Theme precedes Rheme, a boundary with the LH%boundary tone is placed at the boundary between Theme and Rheme.

(62)
```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE maryxml SYSTEM "http://mary.dfki.de/lib/MaryXML.dtd">
<maryxml>
<speaker gender="female">
Der Preis ist <t accent="H*"> dreihundert </t> Euro.
</speaker>
</maryxml>
```

The output of MARY is stored in a MaryXML file where the input above is processed and additional information is automatically added. The output MaryXML file for (62) is represented in (63).

(63)
```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE maryxml SYSTEM "http://mary.dfki.de/lib/MaryXML.dtd">
<maryxml>
<speaker gender="female">
<div>
<phrase>
<t g2p_method="userdict" pos="ART" sampa="'d{6"
syn_attach="1" syn_phrase="NP">
Der
</t>
<t accent="L+H*" g2p_method="lexicon" pos="NN"
sampa="'praIs" syn_attach="0" syn_phrase="NP">
Preis
</t>
<boundary breakindex="3" tone="h-"/>
<t g2p_method="lexicon" pos="VAFIN" sampa="'?Ist"
```

```
syn_attach="1" syn_phrase="_">
ist
</t>
<t accent="H*" g2p_method="lex_compound" pos="VVPP"
sampa="'draI-,hU-N6t" syn_attach="1" syn_phrase="_">
dreihundert
</t>
<t accent="H+L*" g2p_method="rules" pos="NE" sampa="'?OY-ro:"
syn_attach="1" syn_phrase="_">
Euro
</t>
<t pos="$." syn_attach="2" syn_phrase="_">
.
</t>
<boundary breakindex="6" tone="l-l%"/>
</phrase>
</div>
</speaker>
</maryxml>
```

Another example is the utterance in (64:S) represented as the GoDiS internally partitioned string (65) which in turn is represented as the MaryXML marked up text in (66).

(64) U: Wie komme ich von Saarbrücken nach Frankfurt? (How can I get from Saarbrücken to Frankfurt?)

S: Es wäre möglich, von Saarbrücken nach Frankfurt zu FLIEGEN. (It is possible to fly from Saarbrücken to Frankfurt.)

(65)
```
Es waere moeglich, von Saarbruecken nach Frankfurt
zu <RH> <F_RH> fliegen </F_RH> </RH>
```

(66)
```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE maryxml SYSTEM "http://mary.dfki.de/lib/MaryXML.dtd">
<maryxml>
<speaker gender="female">
Es waere moeglich von Saarbruecken nach Frankfurt zu
<t accent="H*"> fliegen </t>.
</speaker>
</maryxml>
```

Again, this partial MaryXML skeleton is processed by MARY and enriched with additional

grammatical and prosodic information.

An example from a different domain is (67:S) which is tagged GoDiS internally as shown in (68) and represented as the SABLE marked up text in (69).

(67) U: Welchen Status hat die Küchenleuchte? (What is the status of the kitchen light?)
S: Die Küchenleuchte ist ANGESCHALTET. (The kitchen light is on).

(68)
```
Die Kuechenleuchte ist <RH> <F_RH> angeschaltet </F_RH> </RH>
```

(69)
```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE maryxml SYSTEM "http://mary.dfki.de/lib/MaryXML.dtd">
<maryxml>
<speaker gender="female">
Die Kuechenleuchte ist <t accent="h*"> angeschaltet </t>.
</speaker>
</maryxml>
```

9.1.2 Varied Synthesized Speech Output in GoDiS Using SABLE

In this section we illustrate the use of the SABLE speech annotation standard to control the intonation of synthesized spoken output of GoDiS, on the basis of the information structure partitioning described in Chapter 6. SABLE is supported by both the MARY and the FESTIVAL TTS systems, but it provides less fine-grained control over intonation than the ToBI-based markup also supported by these systems. We illustrate the conversions from the GoDiS internal information structure annotation to SABLE using English examples for simplicity.

The output of the GoDiS generation module, which is a string annotated with information structure partitioning (Chapter 6), is fed into the GoDiS output module. The GoDiS output module passes it to the SABLE markup language. The information structure partitioning tags are converted into SABLE tags. For instance, for (70:S1), we have the GoDiS internal information structure representation in (71).

(70) U1: How much is the economy flight?
S1: The price is THREE HUNDRED Euro.

(71)
```
The price is <RH> <F_RH> three hundred </F_RH> </RH> euro.
```

In our current experimental version, the internal information structure annotation tags are converted into SABLE tags as follows:

```
<F_RH>  --> <EMPH>
</F_RH> --> </EMPH>
<F_TH>  --> <EMPH>
</F_TH> --> </EMPH>
```

and alternatively,

```
<F_RH> --> <EMPH> <PITCH BASE="+20%">
</F_RH> --> </PITCH> </EMPH>
<F_TH> --> <EMPH> <PITCH BASE="+15%">
</F_TH> --> </PITCH> </EMPH>
```

The result of the conversion is a SABLE marked up text which is saved into a SABLE file. The SABLE marked up text for (71) is represented in (72) below.

(72)
```
<?xml version="1.0"?>
<!DOCTYPE SABLE PUBLIC "-//SABLE//DTD SABLE speech mark up//EN"
      "Sable.v0_2.dtd"
[]>
<SABLE>
<SPEAKER NAME="male1">

The price is <EMPH> three hundred </EMPH>  Euro.

</SPEAKER>
</SABLE>
```

The SABLE file is then sent to FESTIVAL or MARY which output the synthesized speech.

Another possibility to tag Rheme-Focus in SABLE is to use the PITCH tag with which the pitch contour of the utterance can be modified. A version of (72) with modified pitch values added is represented in (73).

(73)
```
<?xml version="1.0"?>
<!DOCTYPE SABLE PUBLIC "-//SABLE//DTD SABLE speech mark up//EN"
```

```
      "Sable.v0-2.dtd"
[]>
<SABLE>
<SPEAKER NAME="male1">

The  price is  <EMPH> <PITCH BASE="+15%"> 300 </PITCH> </EMPH>
euro.

</SPEAKER>
</SABLE>
```

Another example is the utterance in (74:S) represented as the GoDiS internally partitioned string in (75) which in turn is represented as the SABLE marked up text in (76).

(74) U: How can I get from Saarbrücken to Frankfurt?
S: It is possible to FLY from Saarbrücken to Frankfurt.

(75)
```
It is possible to <RH> <F_RH> fly </F_RH> </RH> from Saarbruecken
to Frankfurt.
```

(76)
```
<?xml version="1.0"?>
<!DOCTYPE SABLE PUBLIC "-//SABLE//DTD SABLE speech mark up//EN"
      "Sable.v0_2.dtd"
[]>
<SABLE>
<SPEAKER NAME="male1">

It is possible to <EMPH> fly </EMPH>  from Saarbruecken to Frankfurt.

</SPEAKER>
</SABLE>
```

An example from a different domain is (77:S) which is tagged GoDiS internally as in (78) and represented as the SABLE marked up text in (79).

(77) U: What is the status of the kitchen light?
S: The kitchen light is ON.

(78)
```
The kitchen light is <RH> <F_RH> on </F_RH> </RH>
```

(79)
```
<?xml version="1.0"?>
<!DOCTYPE SABLE PUBLIC "-//SABLE//DTD SABLE speech mark up//EN"
      "Sable.v0_2.dtd"
[]>
<SABLE>
<SPEAKER NAME="male1">

The kitchen light is  <EMPH>   on </EMPH>.

</SPEAKER>
</SABLE>
```

Other utterances may involve both the realization of Rheme-Focus and Theme-Focus. For instance, in (52) repeated below, the system utterance (80:S3) will be partitioned so that the values of the price parameter is assigned Rheme-Focus, and the values of the parameter class will be assigned Theme-Focus.

(80)
S1: Hello, how can I help you?
U1: What is the price of a flight from Paris to London on April fifth?
S2: What class did you have in mind?
U2: I don't know.
S3: BUSINESS class costs ONE THOUSAND EURO.
ECONOMY class costs FIVE HUNDRED EURO.

This partitioning of (80:S3) will be represented in the GoDiS internal information structure annotation as in (81) where the Theme-Focus is only implicitly marked as Focus.

(81)
```
<F> Business </F> class costs <RH> <F_RH> 1000 </F_RH> </RH> euro.
<F> Economy</F> class costs <RH> <F_RH> 500 </F_RH> </RH> euro.
```

If the Theme-Focus is explicitly marked, the GoDiS internal representation will be:

(82)
```
<TH><F_TH>Business</F></TH> class costs <RH><F_RH> 1000</F_RH></RH> euro
<TH><F_TH>Economy</F></TH> class costs <RH><F_RH> 500</F_RH></RH> euro.
```

The so annotated string in (81) will be than converted into the SABLE format represented in (83) where the two foci, (Theme) Focus and Rheme-Focus are labeled with EMPH.

(83)
```
<?xml version="1.0"?>
<!DOCTYPE SABLE PUBLIC "-//SABLE//DTD SABLE speech mark up//EN"
      "Sable.v0_2.dtd"
[]>
<SABLE>
<SPEAKER NAME="male1">

<EMPH> Business </EMPH>  class  costs  <EMPH> 1000 </EMPH> euro.
<EMPH> Economy </EMPH>  class  costs <EMPH> 500 </EMPH> euro.

</SPEAKER>
</SABLE>
```

In order to be able to distinguish between the two foci, we can also modify the pitch contour of the utterance as in (84) where only the Rheme-Focus is emphasized.

(84)
```
<?xml version="1.0"?>
<!DOCTYPE SABLE PUBLIC "-//SABLE//DTD SABLE speech mark up//EN"
      "Sable.v0-2.dtd"
[]>
<SABLE>
<SPEAKER NAME="male1">

 <PITCH BASE="+15%"> Business  </PITCH> class costs
     <EMPH> <PITCH BASE="+20%"> 1000 </PITCH> </EMPH> euro.
 <PITCH BASE="+15%"> Economy  </PITCH> class costs
     <EMPH> <PITCH BASE="+20%">
500 </PITCH> </EMPH> euro.

</SPEAKER>
</SABLE>
```

Additionally, SABLE provides a tag for marking phrase boundaries, BREAK. This tag could be used to mark the boundary between Theme and Rheme in (80). However, the quality of the synthesized speech decreases when BREAK is used.

The implementation we described here is in an experimental phase. The output audio files can be accessed at the project website *http://www.coli-uni-sb.de/cl/projects/siridus.html.*

9.2 Varied Spanish output with Telefónica's TTS

In this section we describe how different possibilities of the Telefónica's TTS system have been used to improve system output by producing varied synthesized speech output.

We also describe some other possibilities the system offers to improve system output, but have not been used within this project.

9.2.1 Telefónica's markups for speech output variation

As we have seen in Section 8.3, this TTS system provides the possibility to label output text to force a specific TTS production or disable the default one. This can be controlled from the dialogue manager by specific markups following the syntax:

```
\MARKUP=PARAMETER\
```

The labels may hold for part of an utterance, a whole utterance or for all following output until the feature is disabled.

Therefore, these labels can be used to vary the default speech output in order to improve the system output.

A way to get this improvement, is to use the labels to control the way some particular type of data are produced. In fact, in dialogue systems, the text that is passed to the TTS is under control and many times the developer knows the type of text that is passed on. In case of a database search system with natural language interface the developer will know the type of the response e.g. a credit card number. Markups give the developer the possibility to decide how these text types is going to be produced, i.e. if the credit card number is going to be read as digits and not as unit as would be the default case for Telefónica's TTS system. Many of the labels are therefore used to disable default treatment for abbreviations, dates, numbers, hours etc. and to activate a certain treatment of these cases.

An example for the Siridus Spanish demonstrator is the treatment of telephone numbers, which can be controlled with the following two labels:

- `\tlf=par\`

Telephone number activated or not, this is to get a number from 4 to 15 digits read as a telephone number. The user can also control how he wants telephone numbers to be read. The following markup

```
\tlf=3d2c2c\
3374612
```

would be read as: three three seven, forty six, twelve.

- `\ltf=par\`

  Telephone numbers with pause limits. This is useful when repeating the numbers a user has said, i.e. together with speech recognition. Some speech recognizers, including Telefónica's one, which is used in Siridus for the Spanish telephone demonstrator, will recognize a number as separate digits with blanks representing the pauses that the user did. Using this mark the TTS pronounces the number as the user did which will facilitate the user's comprehension.

```
\ltf=INI\ 34 1 3 37 41 29 \ltf= FIN\
```

Another way to produce varied system output, is to use the labels to control some intonation parameters.

The following labels are focused on intonation:

- `\pau=par\`

  Pauses. The parameter is in miliseconds and determines the duration of the pause between two words in a text. This can be used to achieve a more natural intonation than the default one or to separate two utterances with more interval.

- `\pit=par\`

  Pitch. This can be used to multiply the basic pitch contour and fundamental voice. The parameter can be set between 64 and 256. A value less than 128 gives a darker voice and a value above 128 a squeakier voice. This enables the user to double the pitch as well as to half it.

- `\spd=par\`

  Controls the speed of the synthesizer. The parameter represents words per minute and may vary between 75 to 300 where the normal value is 150. This can be used to slow down the speech for certain parts, e.g. when pronouncing a telephone number or an e-mail address that the user has to take down.

- `\vol=par\`

  Volume control. This enables the user to control how loud the speaker should talk. If the dialogue system detects a noisy environment it could order the TTS to speak louder.

- `\ppa=par\`

  Deactivate automatic pausing. This will not enhance speech output but the automatic pausing could be deactivated and replaced by manual pausing control.

Combining these labels to control the different parameters, prosody distinctions can be done. For example, an adequate combination of pitch, speed and loudness variation could be used to produce focus on certain words or part of the sentences.

Although some work has been done to test the effects achieved by combining and changing the value of these parameters, very simple labeling has been used in the Siridus Spanish telephone demonstrator, to improve certain output sentences (e.g. changing the default pausing scheme).

However, more elaborated effects could be achieved. In fact, Telefónica has a TTS application with a graphical interface that enables the user to play with pitch, loudness and duration. This can be used to adjust the parameter values to get the desired production. For example, it is possible to adjust the values to get a particular kind of focus. Then, it would be possible to develop an interpreter which would convert markups of a general annotation scheme (such as JSML or SABLE) to the appropriate labels and values for Telefónica's TTS.

9.2.2 Template based phrases

Another way to improve system output intonation with Telefónica's TTS, is to use template based synthesis.

For some commercial systems, where the quality of the speech output is required to be high and where recorded speech cannot been used Telefónica's TTS system gives the developer the choice of using template based phrases.

Many simple telephone applications use just a few messages with a specific structure only varied with output from a database. To improve the prosody of these restricted messages the developer can create template based phrases. The idea is to force a natural prosody taken from a human speaker to the static part of a typical phrase. This gives a more natural sound than normal TTS production but keeps a flexibility that is not possible with

recorded speech. The following two examples illustrate phrases where this method has been used:

- The telephone number of X is Y.
- The train with destination X will depart at Y

The X/Y part use the default TTS prosody , and the rest of the sentence will use the fixed prosody. The text sent to the TTS for the pronunciation of the second phrase would be the following with correspondent marking:

```
\pft=SAL_TREN\Barcelona\sep\12:35\fft\
```

This method has been used in Telefónica for some commercial services. However, it has not been used for the Siridus Spanish demonstrator. The reason for this, is that, although it could be very useful to improve many system outputs, the investment was not worth for a demonstrator. Nevertheless, if it were necessary, this method could be used to improve the system output if it were used commercially.

9.2.3 Other possibilities to improve system output

Apart from the features described before, which are already available for Telefónica's TTS, other possibilities to improve speech output intonation are being explored at the moment in Telefónica I+D.

The main objective of current research is to adapt speech production to dialogue speech. The TTS system is, as many other similar systems, built to handle large texts and read text messages. A lot of effort has been done to develop a system that can handle any given text and, for example, solve abbreviations and treat end of phrases correctly. The use of TTS in dialogue systems is relatively recent, and most TTS systems are not specialized in performing natural dialogue speech for short phrases. This is why current research is focusing on developing a system adapted for normal dialogue speech i.e. spoken language, and not to read up texts i.e. written language.

At this moment, a dialogue-TTS is under development in Telefónica I+D. Its speech output prosody is specially designed for dialogue systems, as it is based on spoken language intonation.

However, this system has not been used for the Spanish demonstrator as some work still needs to be done before it can be used in real systems.

9.3 Experiment Design for Varied Spanish output with FESTIVAL in Delfos 2

In order to improve the quality of synthetic speech, and taking advantage of the information available in a information–state–based architecture, we have designed a set of strategies that will pave the way to more natural synthetic speech.

9.3.1 Selection of Dialogue Prototypes: Base Types

Given the Dialogue Types selected for Siridus, a set of Dialogue Prototypes must be chosen. These prototypes may or not be extracted from a corpus, but it should nonetheless contain a minimum number of dialogue moves. These moves must intuitively go together if the prototypes are artificially created, or must have been extracted directly from a corpus as is, that is, respecting the order and number of moves in the original dialogue. Since it cannot be assumed that this set of common or intuitively normal dialogues covers most of the range of possibilities, other types that may not appear as common or likely should also be consider in order to cover a wider range of cases.

We have initially decided to select a total of six prototypes. Four of them will represent the most likely sets of moves, and the other two will include combinations of moves that may not appear as likely.

The prototypes will be extracted from a corpus if possible. Since corpus availability has not been confirmed yet, artificially generated prototypes are the second, more likely option.

The minimum number of moves for the prototypes has not been decided on yet. However we can presume it should be in the range of 3–6 moves.

9.3.2 Recording and Synthesizing the Dialogues

The next step will be to generate a synthetic and unmodified version of the dialogues with Festival, the synthesizer used for this project. The output will be recorded.

The prototype dialogues will also be scripted and performed by human speakers. Although the conditions under which these recording will take place have not been formally defined yet, the setting would be similar to that of a Wizard of Oz, where the human playing the system would be knowledgeable about the system and its functionality and restrictions.

9.3.3 Comparing Dialogues

Once both sets of recordings are ready, the dialogues will be compared. The prosodic patterns of the synthetic utterances will be compared to those of the natural ones taking into account different factors:

1. Dialogue Type
2. Dialogue Move
3. Dialogue History
4. Information Exchange
5. Expectations
6. Context
7. ...

We will focus on parallelisms and disparities between natural and synthetic utterances and will try to draw patterns based on the influence of the factors above mentioned in the final intonation of the utterances.

9.3.4 Heuristics

Once some relationships have been discerned, a set of rules or heuristics must be created in order to formalize these relationships and incorporate them to the system.

9.3.5 Translating Dialogue Information to Prosodic Labeling

At this point, there should be a set of rules that will help the system decide on what intonation pattern to generate: where to place main and secondary peaks, breaks, etc. The next step is to translate that information to the synthetic speech mark–up language selected, which in this case is SABLE.

9.3.6 Empirical Analysis

In order to test the results, the set of heuristics and its translation to SABLE needs to be implemented into the system. Pre–testing of the SABLE labels in Festival is essential.

Once all the pieces have been put in place, the set of prototype dialogues should be generated under the new rules. Since the translation of heuristics to SABLE may present some unclear cases, more than one version of the prototypes may need to be synthesized.

The original and unmodified synthetic version of the prototypes should be presented to a set of naive subjects, together with the newly synthesized and modified version/s of the same prototypes. It is extremely important that the number of versions presented in the experiment does not undermine the subjects' judgment.

In order to achieve statistical significance, the number of subjects should be of reasonable size. However and given time and resources, a smaller number of subjects would suffice to discern tendencies that would need to be confirm with further experiments in future projects.

9.3.7 Future Work

More than one experiment may need to be carried out in order to achieve the best results and as much information as possible. If the preliminary analysis of the empirical data seems to indicate that a modification in the heuristics or their translation to SABLE may render better results, new sets of prototypes should be generated and presented for naive evaluation.

The results of these experiments may help identify dialogues and specific cases or situations where the default output of the synthesizer can be outperformed by incorporating additional dialogue information.

Unfortunately, the scope of these experiments goes beyond the time frame for this project. The description of these experiments is however valid and useful for the forthcoming work in this area.

Chapter 10

Conclusions

The goal of this report was to explore the use of the information state to control the realization of the system's output. Our main concern was the generation of contextually appropriate variation of prosodic realization.

We substantiated the need to control the prosodic realization of synthesized speech in dialog systems by considering examples where the use of default intonation leads to unnatural or inappropriate out. We also showed that it is not always possible to rely on defaults We discussed several factors involved in producing contextually appropriate synthetic speech, including information structure, dialogue progress history, dialogue move expectations, intelligent barge-in and manipulations of word pronunciation in context. Of these factors, information structure was in the center of our attention in this report.

To establish sufficient theoretical background for our discussion, we provided an overview of the existing approaches to information structure. All the approaches that we considered take information structure as an inherent aspect of utterance meaning, which reflects a partitioning of meaning according to how an utterance relates to the context and how it updates it. From among the many views and terminologies, we settled on the use of Steedman's theory as the most concretely and formally spelled out one today, and also for its concern with prosodic reflexes of information structure. This approach uses two dimensions of information structure, distinguishing between the Theme/Rheme partitioning of the propositional content of an utterance as a whole, which reflects a semantics aboutness relation, and the Background/Focus partitioning within Themes and Rhemes, which serves to indicate how a Theme or a Rheme differs from contextually available similes.

Information structure is a level of meaning which unifies a range of interacting contextually-dependent aspects of utterance realization, including intonation, word order, syntactic constructions, morphological marking, and choice of shortened forms of expressions, such

as anaphora and ellipsis. These realization means are encountered in various combinations within different languages as well as cross-linguistically. We addressed the realization of information structure by means of intonation in English and by means of word order in Czech. We also discussed short utterances from the viewpoint of information structure. This discussion was based on an investigation of short utterances in dialogue corpora, which revealed the prevalence of non-sentential utterances in human-human dialogue, and thus motivated a desire to include these in a dialogue system. Our informal considerations showed that the same underlying abstract representation of Themes, Rhemes and Foci, could in addition to e.g improving system intonation, also be used to handle these short utterances, even though the details remain to be worked out.

However, the task of capturing the full range of interacting means of information structure realization in an integrated fashion in natural language generation (or parsing) is a challenge for contemporary systems. In the remainder of this report we concentratedon the realization of information structure through intonation, because our primary goal in the present work has been to improve the results of synthesized spoken output of a dialogue system. Taking into consideration other means of realization simultaneously remains as a challenge for future development.

Next, we addressed the relation between information structure and context in detail. Building on earlier work in the TRINDI project (Engdahl et al., 2000), we defined a set of rules which specify how an information structure partitioning of utterance meaning can be determined from the information state. Our rules capture the following ideas: the Theme/Rheme partitioning is derived on the basis of the current question under discussion; the Background/Focus partitioning is obtained by comparing the current propositional content with relevant similes, found either in the shared commitments part of the information state or in the representation of the domain. We discussed how these rules can be applied to assign information structure is various cases.

The information state of a system like GoDiS, together with the domain knowledge of particular applications, has proven to be able to accommodate the information structural components and predications of a theory such as Steedman's, which in turn is of course translatable into several other approaches to information structure. This in itself can be seen as a result and as an indication of the viability of our approach, and thus that the combination of information structure theories with an information-state approach to dialogue modeling can lay the foundation for the improvement of system output.

We developed an experimental implementation in the GoDiS system. The rules assigning information structure are implemented as a module that takes as input the propositional content of a dialogue move, and returns this content partitioned according to the current question under discussion, and the contents of the shared commitments and the domain knowledge. The partitioned content produced by this module serves as input to the generation of the surface realization, which produces a string of words along with an annotation

of the information structure partitioning. This annotation uses an internal set of labels. These are subsequently converted into markup suitable for text to speech synthesis.

Finally, we discussed the use of off-the-shelf text to speech synthesis systems for producing contextually varied spoken output in GoDiS, Delfos 2 and the Telefónica dialogue systems. For our experiments with information-structure driven contextual variation of intonation, we have employed the FESTIVAL system for English and the MARY system for both German and English. These systems support not only the SABLE speech markup standard, but also a higher-level annotation of intonation based on ToBI. It is this possibility that makes FESTIVAL and MARY particularly suitable for experimenting with contextually varied intonation. For both systems and the various intonation annotation formats they use we developed interfaces that convert GoDiS output from its internal information structure annotation into input suitable for the individual synthesis modules. Examples of the varied synthesized output can be accessed through *http://www.coli.uni-sb.de/cl/projects/Siridus/*.

We also discussed the production of varied speech output in Spanish using Telefónica's text to speech system. This work did not address the use of information structure, but concentrated on other factors that can improve synthesized speech, and are applicable in contemporary commercial systems. In particular, the use of dialogue context information to properly label the system output used in template-based synthesis has been addressed.

Finally, the impact of varying prosodic realization by taking into account various contextual factors needs to be formally evaluated, to establish whether it actually does lead to improvement of the system behavior, but also to inform further developments. While systematic evaluation has been beyond the scope of the current project, we have presented some general ideas concerning the design of a such experiments, taking into account either existing corpus data or data gathered in controlled setting.

A number of issues remain to be explored with respect to the information structure determining rules. The intricate interplay between information in the information state and in the domain knowledge is one example. In this regard, an area for further exploration is that of taking more of the dialogue history into account. For instance, in our considerations and current implementations of the home device domain the information state will be 'reset' after each action so that if the user first asks the system to, say, switch off the hall light, and the system does so, and the user then asks the system to perform some other action, the system will at that point have no memory, no dialogue history, of the hall light operation. Related is the issue of the interplay between subsequent sentences.

Consider the following example of action-oriented dialogue in the home-device domain:

(85) U: Which devices are on?

S: The kitchen light is on. The kitchen oven is on. ...

If all and only domain knowledge is taken into account for the system's utterance, both instances of *kitchen* will be seen as Focus since they are alternatives of other rooms in the domain, and *light* and *oven* will similarly be Foci. However, if all Foci are intonationally realized, the result will be a system that over-emphasis es. Another possibility is e.g. to just see *oven* as the Focus in the second sentence, in contrast with the *light* in the previous sentence. That is, not to all the time include the whole domain when looking for alternatives, but sometimes to concentrate the contrasts to a smaller context of a preceding or subsequent sentence or utterance. This can also be seen as a problem of which Foci get intonationally realized.

Some of the interplay between dialogue history – when to use the shared commitments – and the domain knowledge, have been discussed in this deliverable, but there is much more work to be done, work that may need user testing or corpus studies. This need is also true of several other issues that can be explored with a view to formulating more detailed rules for information structure determination, and may also lead to the enrichment of the information state.

Another issue open for further investigation and more detailed treatment is that of making more subtle decisions about the realization of information structure by intonation and/or other means, taking into account more information from the information state. So far, the mapping of information structure categories to intonation we are using is very simple, we work with only two types of tune. Discovering systematic relationships between the rich repertoire of intonation patterns and various aspects of the context is subject of ongoing research into the functions of intonation in various languages. The possibility of experimenting with contextually varied intonation in a dialogue system which allow relatively high degree of flexibility can contribute valuable feedback to such investigations.

A closely related issue is that of combining different means of information structure realization within and across languages. There is, however, more work needed to specify the details of the treatment of various realization means and their interaction. We have seen the challenges that arise for example from incorporating word order choices or the the production of short utterances. Taking into account the fine-grained level of distinctions between Rheme-Foci and Rheme-Background components will most certainly involve incorporating more information in the information state.

This line of further development also quite naturally leads to the need of more flexible natural language generation than the currently employed template-based approach allows. Reuable text and sentence generation systems exist, but it is an open question to what extend it is practicable to plug them into a dialogue system without losing efficiency.

Practically oriented goals for further work in the area of text to speech synthesis concern systems which do not yet support intonation annotation standards, but one would want to control their outpur in the ways we discussed in this report. One possibility is to develop interpreters to translate standard annotation schemes (e.g., SABLE, JSML, ToBi...) into labeling interpretable by the individual synthesis systems. This is one of the future goals for Telefnica's TTS. Another Telefnica's future goal is to develop a TTS specifically designed for dialogues.

The need for making more fine-grained semantic choices such as the information-structure partitioning also raises the question of what is a suitable semantic representation formalism. We have so far used a rudimentary formalism where the representation of the meaning of an utterance is closer to the database query contents than to the actual semantics. This, however, obscures many aspects of meaning which are important for more subtle dialoge modelling and management, such as distinguishing between modalities (e.g., wishes/preferences vs. obligations/constraint), degrees of certainty, discourse relations (e.g., correction vs. adding another option), etc. Keeping track of such phenomena requires more elaborate meaning repesentations as well as discourse modelling. Research within the TRINDI ad SIRIDUS projects addressed the use of semantic formalisms which enable partial representations while preserving computational efficiency. More investigation would be needed in this direction, also exploring how information structure can be captured in such formalisms (e.g., indexed semantics, underspecifie discourse representation theory, minimal recursion semantics or hybrid logic).

To summarize, we ascertained that the information state can be used to improve contextual appropriateness of system output. There are many aspects of this research in which we have so far only scratched the surface. Therefore, many issues remain for further investigation

References

Alexander Berman. 2001. Asynchronous feedback and turn-taking. ms.

Dwight Bolinger. 1965. *Forms of English*. Harvard University Press, Cambridge, MA.

C. Brinckmann and J. Trouvain. to appear. The role of duration models and symbolic representation for timing in synthetic speech. *International Journal of Speech Technolgy Research*.

Daniel Büring. 1995. *The 59th Street Bridge Accent: On the Meaning of Topic and Focus*. Ph.D. thesis, Universität Tübingen. Publ. as (Büring, 1997).

Daniel Büring. 1997. *The Meaning of Topic and Focus: The 59th Street Bridge Accent*. Routledge, London.

Daniel Büring. 1999. Topic. In Peter Bosch and Rob van der Sandt, editors, *Focus: Linguistic, Cognitive and Computational Principles*, Natural Language Processing, pages 142–165. Cambridge University Press, Cambridge.

Mary Dalrymple, Stuart Shieber, and Fernando Pereira. 1991. Ellipsis and higher order unification. *Linguistics and Philosophy*, 14.

Elisabet Engdahl, Staffan Larsson, and Stina Ericsson. 2000. Focus-ground articulation and parallelism in a dynamic model of dialogue. Deliverable D4.1, TRINDI.

Staffan Larsson et al. 2002. Flexible dialogue. Deliverable D1.4, SIRIDUS.

Raquel Fernández and Jonathan Ginzburg. 2002. Non-sentential utterances in dialogue: A corpus-based study. In *Proceedings of the Third SIGdial Workshop on Discourse and Dialogue*.

Jan Firbas. 1992. *Functional Sentence Perspective in Written and Spoken Communication*. Studies in English Language. Cambridge University Press, Cambridge.

Claire Gardent and Michael Kohlhase. 1997. Computing parallelism in discourse. In *Proceedings of IJCAI (2)*, pages 1016–1021.

Jonathan Ginzburg. 1996. Interrogatives: Questions, facts and dialogue. In *The Handbook of Contemporary Semantic Theory*. Blackwell, Oxford.

Jonathan Ginzburg. 1999. Semantically-based ellipsis resolution with syntactic presuppositions. In Harry Bunt and Reinhard Muskens, editors, *Computing Meaning*. Kluwer.

Jonathan Ginzburg. forthc. *A Semantics for Interaction in Dialogue*. CSLI Publications and and University of Chicago Press.

Martine Grice, Stefan Baumann, and Ralf Benzmüller. to appear. German intonation in autosegmental-metrical phonology. In Jun Sun-Ah, editor, *Prosodic Typology*. Oxford University Press.

Eva Hajičová and Petr Sgall. 1987. The ordering principle. *Journal of Pragmatics*, 11(4):435–454.

Michael Halliday. 1970a. Language structure and language function. In John Lyons, editor, *New Horizons in Linguistics*, pages 140–165. Penguin, Harmondsworth.

Michael A.K. Halliday. 1970b. *A Course in Spoken English: Intonation*. Oxford Uniersity Press, Oxford.

Michael A.K. Halliday. 1985. *Introduction to Functional Grammar*. Edward Arnold, London, U.K.

Jim Hieronymus, Stina Ericsson, and Staffan Larsson. 2002. Associating the dialogue move engine with speech output. Deliverable D2.2, SIRIDUS.

Julia Hirschberg and Janet Pierrehumbert. 1986. The intonational structuring of discourse. In *Proceedings of the Annual Meeting of the Association of Computational Linguistics*, pages 136–144. ACL.

Gérard Huet. 1975. A unification algorithm for typed λ-calculus. *Theoretical Computer Science*.

Lauri Karttunen and Stanley Peters. 1979. Conventional implicature. In Choon-Kyu Oh and David A. Dinneen, editors, *Syntax and Semantics: Presupposition*, volume 11, pages 1–56. Academic Press.

Ivana Kruijff-Korbayová, John Bateman, and Geert-Jan M. Kruijff. 2002a. Generation of contextually appropriate word order. In Kees van Deemter and Rodger Kibble, editors, *Information Sharing*, Lecture Notes, pages 193–222. CSLI.

Ivana Kruijff-Korbayová, Elena Karagjosova, and Staffan Larsson. 2002b. Enhancing collaboration with conditional responses in Information Seeking Dialogues. In Johan Bos, Mary Ellen Foster, and Collin Matheson, editors, *Proceedings of the 6th Workshop on the Smeantics and Pragmatics of dialogue*, pages 93–100. The University of Edinburgh, 4–6 September 2002.

Ivana Kruijff-Korbayová. 1998. *The Dynamic Potential of Topic and Focus: A Praguian Discourse Representation Theory*. unpublished Ph.D. thesis, Faculty of Mathematics and Physics, Charles University, Prague, Czech Republic.

Geert-Jan M. Kruijff. 2001. *A Categorial-Modal Logical Architecture of Informativity: Dependency Grammar Logic & Information Structure*. Ph.d. dissertation, Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University, Prague, Czech Republic.

Knud Lambrecht. 1994. *Information Structure and Sentence Form: Topic, Focus, and the Mental Representations of Discourse Referents*. Cambridge University Press, Cambridge.

Staffan Larsson, Gabriel Amores, Rebecca Jonsson, and José Quesada. 2002. Siridus system architecture and interface report (enhanced version). Deliverable D6.3, SIRIDUS.

Staffan Larsson. 2002. *Issue-based Dialogue Management*. Ph.D. thesis, Göteborg University.

Vilém Mathesius. 1975. On information bearing structure of the sentence. In S. Kuno, editor, *Harvard studies in syntax and semantics*. Harvard University Press.

Janet Pierrehumbert. 1980. *The Phonology and Phonetics of English Intonation*. Ph.D. thesis, MIT. Distributed by Indiana University Linguistics Club, Bloomington.

Massimo Poesio, Stephen Isard, Helen Wright, James Hieronymus, Robin Cooper, Staffan Larsson, and Johan Bos. 2000. Prosodic cues for information structure. Deliverable D4.1, TRINDI.

Pilar Manchón Portillo. 1999. *Psychokinetically Inhibited Manoeuvrability: Towards More Natural Synthetic Speech*. Msc thesis, Edinburgh University.

Scott Prevost and Mark Steedman. 1994. Specifying intonation from context for speech synthesis. *Speech Communication*, 15:139–153.
Scott Prevost. 1995. *A Semantics of Contrast and Information Structure for Specifying Intonation in Spoken Language Generation.* Ph.D. dissertation, IRCS TR 96-01, University of Pennsylvania, Philadelphia.
Stephen G. Pulman. 1997. Higher order unification and the interpretation of focus. *Linguistics and Philosophy*, 20:73–115.
Mats Rooth. 1985a. *Association with Focus.* Ph.D. thesis, University of Massachusetts, Amherst.
Mats Rooth. 1985b. *A Theory of Focus Interpretation.* Ph.D. thesis, Graduate School of the University of Massachusetts, Amherst, Massachusetts.
Mats Rooth. 1992a. A theory of focus interpretation. *Natural Language Semantics*, 1:75–116.
Mats Rooth. 1992b. A theory of focus interpretation. *Natural Language Semantics*, 1:75–116.
Marc Schröder and Jürgen Trouvain. 2001. The german text-to-speech synthesis system MARY: A tool for research, development and teaching. In *The Proceedings of the 4th ISCA Workshop on Speech Synthesis, Blair Atholl, Scotland.*
Petr Sgall, Eva Hajičová, and Jarmila Panevová. 1986. *The meaning of the sentence in its semantic and pragmatic aspects.* Reidel, Dordrecht, The Netherlands.
Mark Steedman and Ivana Kruijff-Korbayová. 2001. Introduction: Two dimensions of information structure in relation to discourse structure and discourse semantics. In Ivana Kruijff-Korbayová and Mark Steedman, editors, *Information Structure, Discourse Structure and Discourse Semantics, ESSLLI2001 Workshop Proceedings*, pages 1–6, Helsinki, Finland, August 20-24. European Summer School in Logic, Language and Information (ESSLLI), The University of Helsinki. `http://www.coli.uni-sb.de/~korbay/esslli01-wsh/Proceedings/intro-text.ps.gz`.
Mark Steedman. 1996. *Surface Structure and Interpretation.* M.I.T. Press, Cambridge, MA.
Mark Steedman. 2000a. Information structure and the syntax-phonology interface. *Linguistic Inquiry*, 31(4):649–689.
Mark Steedman. 2000b. *The Syntactic Process.* M.I.T. Press, Cambridge, MA.
Enric Vallduví. 1992. *The Informational Component.* Garland, New York.