

Response Planning in Information-Seeking Dialogues

Kristiina Jokinen

A thesis submitted to the University of Manchester for the degree of
Doctor of Philosophy in the Faculty of Technology

1994

Centre for Computational Linguistics
University of Manchester Institute of Science and Technology

Abstract

The thesis specifies requirements for robust, cooperative and coherent dialogue systems and advocates a new approach, Constructive Dialogue Management. A prototype of a system capable of dealing with the planning of system responses in factual information seeking dialogues is presented.

Constructive Dialogue Management is based on the theoretical foundation of Communicative Activity Analysis (CAA), a pragmatic theory of communication developed by Jens Allwood, and on the empirical experience gained in the PLUS project which aimed to improve robustness in dialogue systems by a pragmatics-based approach.

In CAA, communication is regarded as cooperative activity between rational agents, constrained by the requirements of ideal cooperation: the communicators have a joint purpose, they obey communicative obligations by showing cognitive and ethical consideration in their contributions and they trust the partner to behave in a rational way.

In Constructive Dialogue Management, the use of contextual knowledge is widened to include the enablements and requirements for communication. As a result, the negotiative nature of dialogues can be managed locally, and both predefined dialogue grammar and speech-act classification can be abandoned. Furthermore, by refining the goal with the help of communicative knowledge about ideal cooperation and rational activity, boundary problems between planning and realisation can be overcome.

Within this framework, solutions to two significant subproblems of response planning are presented: how to guarantee thematic coherence with respect to the preceding context, and how to combine appropriate explicitness with elliptical generation based on the communicative principles of Constructive Dialogue Management.

Declaration

No portion of the work referred to in this thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institution of learning.

Acknowledgements

Many friends and colleagues provided me with help in various stages of the work, and I wish to express my warmest thanks for their support.

I would especially like to thank my supervisor William J. Black, not only for his supervision and comments on the implementation of the system, but also for providing the opportunity to work in the PLUS project, whose ambitious atmosphere shaped the basic ideas of the thesis.

I am also grateful to Jens Allwood for his insightful comments on the theoretical part of the thesis and for several stimulating discussions on Communicative Activity Analysis.

Melpomeni Alexa, Jean Carletta and Evelyn van de Veen gave me constructive comments on the draft versions of the thesis, and Joakim Nivre helped me especially with discussions about cooperativeness and robustness. Marek Hatala and Jose Herrero collaborated with the implementation of the ideas in the PLUS project, Ahmed Gues-soum helped with the knowledge base update procedures and Jun-ichi Fukumoto with UNIX. Nancy Underwood was a cooperative colleague and friend throughout the PLUS project and the writing of the thesis. I would also like to thank all the partners in the PLUS project: they have directly or indirectly influenced the work presented in the thesis. Special thanks go to Graham Wilcock who was always able to find the lost goals for me with his support and encouragement.

Finally, I would like to thank my mother Tuula Jokinen for her continuous long-distance support. I dedicate the thesis to my late father Valde Jokinen, who taught me to think through.

Manchester, July 1994

Kristiina Jokinen

Contents

1	Introduction	1
1.1	Goals and objectives	1
1.2	The Structure of the Thesis	7
I	Background	11
2	The PLUS project	12
2.1	Aims of the project	12
2.2	Overview of the system	13
2.3	Dialogue corpus	16
2.3.1	General set-up for corpus collection	17
2.3.2	Nature and type of dialogues	19
2.3.3	Consequences for dialogue management	24
2.4	Evaluation	25
3	Theoretical background	27
3.1	Pragmatics and natural language processing	27
3.1.1	What is pragmatics?	28
3.1.2	Cooperation	32
3.1.3	Coherence	35

3.1.4	Robustness	39
3.2	Approaches to dialogue management	40
3.2.1	The Intention-based approach	42
3.2.2	The Structural approach	47
3.2.3	Conversation Analysis	51
3.2.4	Evaluation of the approaches	52
3.3	Natural language generation	54
3.3.1	Introduction	54
3.3.2	Cooperativeness in Response Planning	57
3.4	Conclusion	58
4	Communicative Activity Analysis	60
4.1	Levels of Communication	61
4.2	Rational AgenthooD	63
4.3	Ideal Cooperation	65
4.4	Full-blown Communication	66
4.5	Communicative responsiveness	67
4.6	Roles and activities of participants	69
4.7	Expressive and evocative dimensions of communication	72
4.8	Conclusion	73
II	Pragmatics-based Dialogue Management	75
5	Constructive Dialogue Management	76
5.1	System robustness and dialogue studies	77
5.1.1	Physical requirements for the user interface	77

5.1.2	Efficiency	78
5.1.3	Syntactically problematic input and output	79
5.1.4	Speaker initiatives	80
5.1.5	Conclusion	86
5.2	Desiderata for dialogue systems	88
5.2.1	Dialogue coherence	89
5.2.2	System initiatives	90
5.2.3	Joint Purpose	94
5.2.4	Obligations	107
5.2.5	Interactive communication management	109
5.2.6	Conflicts	111
5.3	Conclusion	115
6	Design of a Constructive Dialogue Manager	117
6.1	Dialogue manager architecture	118
6.1.1	Overview	118
6.1.2	Context Model	120
6.1.3	Concepts and units	121
6.2	Sketch of Constructive Dialogue Management	134
6.3	Pragmatics-based planning	136
6.3.1	The generation task	136
6.3.2	Planning algorithm	137
6.4	Cooperative system responses	139
6.4.1	Cooperativeness and presuppositions	139
6.4.2	Cooperativeness and plans	141
6.4.3	Cooperativeness and false implicatures	142

6.5	Conclusion	144
7	Dialogue Coherence	146
7.1	Terminological distinctions	148
7.1.1	Topic and focus	149
7.1.2	Central Concept and NewInfo	152
7.2	Topic shifting	154
7.2.1	World and application knowledge	154
7.2.2	Topic shifting and rational agents	155
7.2.3	Topic shift rules	159
7.2.4	Corpus examples	163
7.3	Determining Central Concept	166
7.4	Conclusion	170
8	Explicitness and Ellipsis	172
8.1	Explicitness in generation	173
8.2	Explicitness and cooperative communication	175
8.2.1	Responsiveness and Minimalism Principles	175
8.2.2	Relevance criteria	178
8.3	Ellipsis	183
8.3.1	Definitions	183
8.3.2	Realisable concepts	185
8.4	Examples	187
8.5	Conclusion	194
9	Implementation	195
9.1	Data structures	195

9.2	Control structure	199
9.3	Acceptance of the user contribution	202
9.3.1	Parsing	203
9.4	Interpretation of the user contribution	205
9.4.1	Translation of the semantic representation	205
9.4.2	Recognition of the user goal	207
9.4.3	User obligations	212
9.5	Reaction to the user contribution	212
9.5.1	Evaluation of the user contribution	213
9.5.2	Responding to the user contribution	229
9.5.3	The system's obligations	230
9.5.4	Response planning	235
9.5.5	Surface generation	244
9.6	Conclusion	245
10	Discussion	246
10.1	Conclusions	246
10.2	What theoretical ideas give the system its power	250
10.3	Future Directions	251
	Bibliography	252
	Appendix	268

List of Figures

2.1	The PLUS System.	14
3.1	Elements of the communicational content of an utterance	29
3.2	Robustness features in PLUS	41
4.1	The system's communicative obligations and rights	71
4.2	Expressive and evocative dimensions of four communicative acts	73
5.1	Relations between sequential contributions and dialogue contexts. . . .	91
5.2	Joint purposes for the system	98
5.3	Joint purposes for the system (2)	108
6.1	The architecture of the system.	118
6.2	Information in Contextual Knowledge Base.	121
6.3	Representation of the context after the first user contribution <i>I need a car.</i>	122
6.4	Joint purposes.	125
6.5	The relative 'scopes' of grammatical units, communicative acts and contributions.	127
6.6	Conventional expressive and evocative attitudes	129
6.7	Expressive, evocative and evoked attitudes of the user contribution <i>I need a car.</i>	131

6.8	Expressive, evocative and evoked attitudes of the user contribution <i>In Entwistle.</i>	132
6.9	Expressive, evocative and evoked attitudes of the user contribution <i>Are there any car hire companies in Bolton?</i>	132
6.10	The different types of discourse referents.	134
7.1	Part of the world model.	154
7.2	Focussed concepts in the discourse about John	160
7.3	Coherent topic shifts	162
7.4	The topic tree for sample dialogue 55	165
7.5	The topic tree for sample dialogue 57	167
8.1	Mapping types between world model concepts, semantic predicates and lexemes	185
8.2	Expressive, evocative and evoked attitudes of the user contribution <i>I need a car.</i>	190
8.3	Expressive, evocative and evoked attitudes of the user contribution <i>I want a car.</i>	191
8.4	Expressive, evocative and evoked attitudes of the user contribution <i>Are there any car hire companies in Bolton?</i>	191
8.5	Expressive, evocative and evoked attitudes of the user contribution <i>How many car hire companies are there in Bolton?</i>	192
9.1	Data flow in the system.	200
9.2	The control flow of the system's reaction.	214

Chapter 1

Introduction

1.1 Goals and objectives

Consider the following dialogue between a user and a system which provides information from the Yellow Pages. The dialogue is a sample dialogue used in the PLUS project (*Pragmatics-based Language Understanding System*, ESPRIT II project No. 5254).

- (1)
- (1) Service: Welcome to the Yellow Pages information service.
Please state your query.
 - (2) User: I need a car.
 - (3) Service: Do you want to buy or rent one?
 - (4) User: Rent.
 - (5) Service: Where?
 - (6) User: In Atalante.
 - (7) Service: Where is Atalante?
 - (8) User: In Rennes.
 - (9) Service: Ok. The following car hire companies are in Rennes:
<list of car hire companies>
 - (10) User: What is the cheapest car hire company?
 - (11) Service: Sorry. There is no information on prices.
Please contact the company.
 - (12) User: Ok, thanks. Bye.
 - (13) Service: Thanks for calling. Bye.

The dialogue exemplifies several different aspects of human-computer interaction to

which a robust dialogue manager must attend (numbers refer to the dialogue contributions):

- The dialogue begins with a vague request and this is refined until it matches some part of the information in the Yellow Pages database (1-9).
- World knowledge is used to interpret the concepts and their relations: e.g. that ‘needing’ means ‘wanting to have’ and ‘cheapest company’ refers to ‘prices’ (3,10).
- Knowledge about the activity (application) is used to match the user request with the tasks that the system is capable of performing: the user’s request for a car is connected to the system’s task to give information about car hire companies (2-9).
- Clarification about unknown concepts is initiated (7).
- Helpful re-routing information is given when the system is unable to comply with the request (11).
- Consecutive queries about location are phrased differently (5,7).
- The user may continue with follow-up questions (10).
- Both the user and the system contributions can be elliptical (4,5,6,8).
- Anaphoric pronouns like ‘one’ are used to refer to previous discourse referents, and thus to tie the discourse parts together (3).
- A pragmatic marker ‘ok’ is used to give feedback about the acceptance of the previous response and topic change (9).
- Conventional markers of politeness ‘sorry’, ‘thanks’ are used (11,12).
- Boundary exchanges (closing) are used to mark the end of the interaction (12-13).

These aspects provide an empirical basis for system design as well as a theoretical challenge for modelling cooperativeness in dialogue management. However, to improve the flexibility and naturalness of human-computer interfaces, the system needs to have extensive knowledge not only about the syntax and semantics of the language (in our case: English), but also about the world and the particular application, about topic and focus, dialogue management, cooperativeness and communicative appropriateness. In other words, the system needs to exploit *pragmatic* knowledge in its interpretation and planning processes, and it needs to restrict its reasoning to *contextually relevant* information.

The work described in this thesis started within the ESPRIT project PLUS whose goal was to build a robust, flexible interface to electronic Yellow Pages. The project studied the role of context and context-based reasoning in human-computer interaction and the guiding principle was the use of pragmatic knowledge and pragmatic reasoning to achieve robustness. The aspects listed above were some of the issues that the project tackled with the help of contextual and pragmatic reasoning.

This thesis is an extension and elaboration of this kind of pragmatics-based approach to dialogue management. It concerns the desiderata for a cooperative and flexible dialogue system, and presents a prototype of a system that addresses robustness from the point of view of communicative competence. The new approach to model information processing in dialogue systems is called *Constructive Dialogue Management*. The thesis deals especially with the planning and generation of system responses, and the communicative principles that govern these processes.

Theoretical and empirical basis

The *theoretical* basis of the thesis can be found in the concepts of ideal cooperation and rational, motivated agenthood as developed by Jens Allwood. We call the theoretical basis of our work *Communicative Activity Analysis*, *CAA*, as it was called in the PLUS context. However, it should be emphasised that Allwood's ideas have functioned as a source of inspiration rather than a fully developed formal basis for our work. The work done in the thesis is original in that it takes the CAA principles and tries to base desiderata for cooperative and robust dialogue management on a unified pragmatic theory. It is also the first attempt to specify, formalise and implement Allwood's ideas for a prototype dialogue system, loosely following the lines sketched in Allwood and Haglund (1991). The work also uses other insights in cooperative dialogue management and response generation research in general, most notably Joshi et al. (1984), Moore and Swartout (1990), Galliers (1989) and McCoy and Cheng (1990).

Following CAA, communication is understood as cooperative activity between rational agents. The agents analyse and evaluate the partner's contributions and report back the

result of this evaluation. However, they do not use language only to obtain their own goals in a one-way activity, but are engaged in a joint enterprise: while elaborating her own goal, the speaker must indicate that the response also addresses the expectations which the partner meant to express and evoke by her contribution. This is so because communication creates social, normative obligations which oblige the agents to act according to the principles of ‘ideal cooperation’, basically that they are willing to continue the dialogue until both have achieved their goals and both are satisfied with the result. Of course, human interaction also involves conflicts, but as claimed in CAA, if the conflict becomes so serious that it makes any cooperation impossible, communication will eventually break down as well.

The *empirical* basis of the thesis is grounded in the corpus collection work conducted in PLUS. The aim in this thesis is not to model naturally occurring dialogues as such, but rather, to study what kind of phenomena occur in information seeking dialogues between a human and a computer, and then, based on the communicative insights of CAA, design a system that is able to respond, in an adequate and robust way, to these phenomena. This requires idealisation and generalisation over the empirically obtained dialogue sample. Many issues discussed in conversation analysis studies (e.g. pauses, overlaps, external interruptions, multi-partner conversations, influence of social context and roles) are not considered in the thesis. This is not because of their irrelevance to a robust dialogue manager or to CAA, but simply because their study is beyond the chosen scope and limits of this thesis.

Goals and research objectives

The goals and research objective of the thesis can be summarised as follows:

- To study the use of pragmatics and principles of communication to improve robustness in dialogue management.
- To use Allwood’s ideas about ideal cooperation and rational motivated activity in achieving robust, flexible dialogue management.

- To explore the scope of the ideas in response planning, especially in determining thematic coherence and elliding material.
- To give a prototype implementation of the ideas of Constructive Dialogue Management.

The thesis will demonstrate CAA in dialogue management, especially as a basis for strategies for deciding the appropriate continuation, or the joint purpose of the dialogue. It will also discuss thematic coherence of the dialogue in terms of ‘Central Concept’ and ‘NewInfo’, and the relevance and appropriateness of the surface form of the planned response. Topic determination and ellipsis generation can both be seen as signs of the system’s compliance with the principles of ideal cooperation and rational action: show implicitly how the system has understood the user’s contribution and the goals set for the dialogue, and thus exemplify the notion of ‘communicative competence’ of the system.

Original contribution

The original contribution of the thesis deals with the identification of robustness features of information seeking dialogues, and the formalisation of these features as constraints on dialogue management. Allwood’s ideas of communication as rational cooperative action are applied to human-computer interaction, and a unified basis for robust and cooperative dialogue manager is sketched with the help of Communicative Activity Analysis. An implementation of a prototype system based on these principles is also presented.

The system adheres to dialogue management which is called *constructive*: dialogue is a joint task whereby the partners construct the model of the common purpose. This is based on the negotiative nature of dialogues: the participants try to achieve their own goals in the cooperative activity by means of communicating and negotiating about their wants and wishes. This requires knowledge about the world and the particular dialogue situation, but also about speakers’ role and communicative obligations. A

novel feature in our system is that it does not have a dialogue grammar, but uses the contextual knowledge base and its updates as the control mechanism for the appropriate responses. The context includes the expressive and evocative attitudes of the previous contributions, the speakers' goals, new information conveyed by the contributions and discourse referents with a distinguished Central Concept. Each contribution is evaluated with respect to the context, and the response is built as a reaction to the changed context in accordance with communicative obligations.

Besides constructive dialogue management with context updating, a novel idea is also that response planning is based on the new information that is to be communicated. The default surface realisation is thus an elliptical contribution consisting of the new information alone, which is augmented to a full sentence only if it does not fulfil the four relevance criteria. The relevance criteria are stated in terms of the expressive and evocative dimensions of the contribution.

The system presented in the thesis makes a new approach to the distinction between planning ('what to say') and realisation ('how to say'), and attempts to overcome the boundary problems between them by a pragmatics-based approach. The task for generation is to define how surface level utterances are planned in the context so that the contribution communicates the speaker's goal and is coherent and consistent with the context. Given the communicative goal of the system, the planner proceeds in refining the goal step-wise with respect to application knowledge, contextual information and surface realisation criteria, complying with the obligations expressed and evoked by the preceding utterances. Planning obeys two general principles, based on the Allwoodian concept of 'ideal cooperation': the Responsiveness Principle ('report the new information that results from the evaluation of the partner's contribution') and the Minimalism Principle ('say only as much as necessary to communicate the goal'). A fully determined semantic representation is passed over to the surface generator.

1.2 The Structure of the Thesis

The thesis has two parts. Part I is devoted to the theoretical and empirical background of the research, and Part II describes new work on dialogue management and response planning. The goals and objectives of the research are given in Chapter 1. The emphasis of the thesis is on new theoretical insights in dialogue management and response planning.

Part I has three chapters corresponding to the practical and theoretical context of our research.

Chapter 2 describes the PLUS project within which the early part of the research was carried out. The ambitious goal of PLUS, to build a robust dialogue manager whose functionality is based on pragmatic and contextual reasoning, also frames our research objectives. The chapter briefly presents the aims of the project and gives an overview of the PLUS system. Also the corpus collection work conducted in PLUS is discussed, since this forms the empirical basis for the research described in this thesis.

Chapter 3 presents the theoretical basis of the research. The three sections on pragmatics, dialogue management and natural language generation form the three main topics of the thesis. We first discuss pragmatics in natural language processing in general, and address especially the problems of cooperativeness, coherence and robustness. These are discussed from the viewpoint of pragmatics-based dialogue management, as background for our argument in Part II that a uniform pragmatic theory of communication is needed for truly robust dialogue management. We then compare two different approaches to dialogue organisation, the intention-based approach and structural approach, and discuss their suitability for robust dialogue management. Finally, we discuss the task of natural language generation and survey problems in planning and generation pertinent to the thesis.

Chapter 4 introduces Communicative Activity Analysis (CAA) as formulated in Allwood (1976). CAA serves as the theoretical basis for our research in applying communicative principles to dialogue management. Some of CAA ideas were also included

in PLUS. Here we discuss the main concepts of the theory in relation to general communication. These concepts are further interpreted for the research purposes of this thesis in Chapter 5.

Part II deals with our own work. The six sections describe an approach to dialogue management which we call ‘Constructive Dialogue Management’ and base on the negotiative nature of dialogues and the communicative competence of the participants as rational, cooperative agents. The design of a dialogue system that implements some of the pragmatic principles of CAA, concentrating especially on the determination of the system goal, thematic coherence and ellipsis generation, is presented.

Chapter 5 presents the view point taken in the thesis with respect to the issues given in the theoretical background chapters. We first discuss robustness on the basis of our corpus studies and identify four general requirements for a robust dialogue management: physical feasibility of the interface, efficiency of reasoning components, ability to cope with syntactically problematic input and ability to initiate responses which clarify vague, misunderstood or otherwise partial input. We concentrate especially on the last requirement, and develop the ‘Constructive Dialogue Management’ approach to dialogue management on the basis of empirical data and CAA principles of ideal cooperation and rational agenthood.

Chapter 6 gives the architectural perspective to dialogue management and presents the design of our dialogue system. In Constructive Dialogue Management, the use of contextual knowledge is widened to include the enablements and requirements for communication. As a result, the negotiative nature of dialogues can be managed locally, and both predefined dialogue grammar and speech-act classification can be abandoned. Furthermore, by refining the goal with the help of communicative knowledge about ideal cooperation and rational activity, boundary problems between planning and realisation can be overcome. We also study pragmatics-based generation and describe the response planner and the planning algorithm. The system refers to the ‘no-expression, specification-less’ view of generation advocated by McDonald (1993): the information content of the system response is expressed in non-linguistic terms and gradually specified in the course of generation process. Cooperative system responses

are also discussed and compared to the requirements found in the relevant literature (Kaplan’s COOP, Wahlster et al.’s HAM-ANS, Joshi et al.’s (1984) work).

Chapter 7 discusses the thematic coherence of dialogues in the Constructive Dialogue Management framework. We introduce the distinction between aboutness (what the dialogue is about and what is in the background) and information status (new vs. old information), and accordingly, define the notions of ‘Central Concept’ and ‘NewInfo’. Thematic coherence is traced with the help of Central Concept and topic shifting rules which are based on world model relations. The claim is that dialogue coherence in terms of identifiable links between discourse referents is an expression of the fact that the higher level communicative obligations have been appropriately addressed.

Chapter 8 concentrates on the generation of elliptic contributions and especially, on combining ellipses with communicationally appropriate explicitness. We base the discussion on two principles that describe the speakers’ communicative competence: the Responsiveness Principle and Minimalism Principle, and describe how system responses are planned so that a fairly specified semantic representation is given to the surface generator to realise. The default system contribution is an elliptic contribution which only conveys the NewInfo, and this be augmented to a full sentence contribution given that the four relevance criteria are not fulfilled. The relevance criteria are: Accuracy, Validity, Consistency, and Freedom-From-False-Implicatures.

Chapter 9 is devoted to the actual implementation. We give an overview of a prototype system which addresses robustness and cooperativeness requirements by implementing the CAA principles of ideal cooperation, joint purpose, communicative obligations, and expressive and evocative dimensions of contributions. The system also reasons on the thematic coherence of contributions and determines ellipsis. Since the thesis has not concentrated on the analysis side of dialogue management nor knowledge representation issues, this chapter is not intended to present a full computational implementation of the ideas: the system also needs a shallow natural language front end and a comprehensive world model component. Rather, the implementation aims to show that the ideas presented in the thesis are sound for building a full scale dialogue manager.

Chapter 10 contains discussion and evaluation of the system, and also points to further work.

Part I

Background

Chapter 2

The PLUS project

This chapter gives a brief overview of the PLUS project *Pragmatics-based Language Understanding System*, ESPRIT II project No. 5254, within which our research started.¹ We also review the corpus collection work conducted at the beginning of the project which forms the empirical basis for our research. Desiderata for a robust dialogue system are drawn in Section 5.1 on the basis of collected sample dialogues.

2.1 Aims of the project

The aims of the project were to study the use of context and pragmatics in human-computer dialogues and to build a robust and cooperative interface to an electronic Yellow Pages based on pragmatic and contextual reasoning. The keynote of the project was “to achieve robustness by treating natural language as an activity whose essential characteristics is to convey meaning that is both appropriate and relevant in the context” (Black et al., 1991). Robustness was understood as the system’s ability to behave adequately in a wide range of situations: it should be capable of dealing with extra-grammatical input (elliptical fragments and misspellings), and flexible enough to allow a real dialogue with the user. The project contrasted itself to ‘low-level’ approaches

¹We gratefully acknowledge the financial support of CEC in the early stage of our research.

which rely on the correction of errors at the level where they are spotted without reference to the contextual knowledge of the dialogue, thus constraining the user to stick to the accepted syntax, and missing the heart of the problem which is to react to the user's message appropriately in the context. To demonstrate the capabilities of the system on a realistically sized application, PLUS chose an interactive Yellow Pages Information Service as the demonstrator.

The participating organizations were: CAP GEMINI INNOVATION, Paris (the coordinating contractor); ITK, Tilburg; Omega Generation, Bologna; CAP GEMINI SCS BeCom GmbH, Hamburg; LIMSI, Paris; UMIST, Manchester; the University of Bristol; and the University of Göteborg. The project started in November 1990 and ended prematurely in June 1993.

2.2 Overview of the system

The PLUS system consisted of the Natural Language Engine (parser and surface generator), the Dialogue Manager and the Knowledge Bases. The Dialogue Manager was composed of three principal subcomponents which were: the *Cognitive Analyser (CA)*, the *Goal Formulator (GF)* and the *Response Planner (RP)*. The Contextual Knowledge Bases comprised the following: the Discourse Model, Pragmatic Rules, World Model and Yellow Pages database. Figure 2.1 illustrates the PLUS system and the internal composition of the Dialogue Manager at a conceptual level.

The task of the Natural Language Engine was to provide an interface between the user and the Dialogue Manager. PLUS was interested in grammar development to the extent that it serves pragmatics. Thus the parser was not to fail, but to produce as full a semantic representation as possible for any utterance presented to it. The surface generator was to get a fully specified semantic representation of the system response and translate this into a string of word forms. The semantic representation was expressed in a quasi-logical form which contained grammatical information needed to interpret or generate an utterance. The parser used unification-based HPSG grammar, while

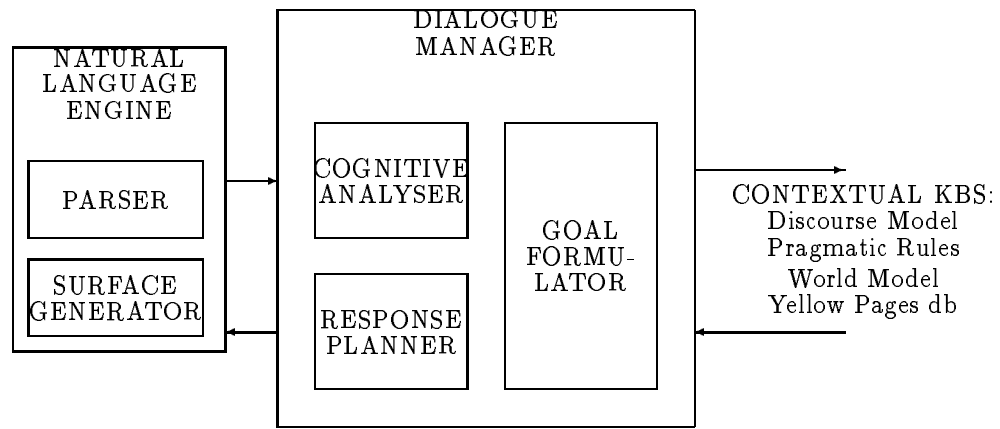


Figure 2.1: The PLUS System.

the generator was a bottom-up generator based on categorial grammar and extended with features. They are documented in Rentier et al. (1993).

The Cognitive Analyser took care of plan recognition whereby the system found the user's communicative goal, and the Goal Formulator decided which goal it should try to achieve as a cooperative and appropriate response to the user's input. The task of the Response Planner was to plan a quasi-logical representation for the next system response, starting from the goal formulated by the Goal Formulator. This task was divided into subtasks that dealt with:

- finding appropriate dialogue acts,
- deciding on an appropriate topic in the context,
- planning a full sentence or an elliptical sentence,
- deciding on appropriate referring expressions
- choosing between ambiguous quasi-logical predicates.

The PLUS discourse model employed two complementary pragmatic theories: Communicative Activity Analysis (Allwood, 1976, 1978, 1992; Allwood & Haglund, 1991)

and Dynamic Interpretation Theory (Bunt, 1990, 1991). The former emphasises communication as rational cooperative activity, and the latter concentrates on dynamic aspects of dialogue interpretation, regarding communicative acts as functions from one dialogue state to another. Communicative Activity Analysis is the subject of Chapter 4, and Dynamic Interpretation Theory is discussed in Section 3.2.1.

In PLUS, dialogues were understood as linguistically expressed complexes of communicative actions which can be described in terms of their effects on the context. The Discourse Model contained all the dynamic knowledge necessary to model the dialogue, and the dynamic interpretation of the dialogue was recorded chronologically by the dialogue history. PLUS also used dialogue structure rules which will be discussed in Section 3.2.2. The PLUS Discourse Model is documented in the project deliverable Jokinen et al. (1992).

The system's reasoning was based on pragmatic knowledge and world knowledge. The PLUS World Model is described in Cavalli et al. (1992b). Pragmatic rules encoded the system's knowledge about cooperative, rational behaviour in information seeking situations as defined in Allwood and Bunt (1992). The rules are described in Bego et al. (1992).

Pragmatic rules were expressed as declarative constraints and were of two types: *pragmatic constraints* dealt with the consistency of the Discourse Model, and *pragmatic rules proper* defined pragmatic actions to manipulate the Discourse Model. Pragmatic competence of the PLUS system dealt with

- roles and expectations of the speakers,
- communicative functions and their realisation in dialogue acts,
- topic,
- reference,
- explicitness and implicitness,
- dialogue structure,
- feedback.

The rules were not divided according to their function, but rather, according to the contextual information that they encoded. As the context was represented in a declarative way, the same knowledge could be used in different operations.

The context-based reasoning required facilities that supported the use of complex knowledge bases and efficient reasoning. PLUS was also concerned with knowledge base management and explored issues in language-independent knowledge representation and knowledge base update techniques.

As part of the Knowledge Base Management System of PLUS, the knowledge base update procedure developed by Guessoum and Lloyd (1990, 1991) was used for experimenting with knowledge base interaction through updates. In this procedure, briefly, when a request to update the knowledge base comes, and the update leads to a violation of the knowledge base's integrity constraints, the update procedure produces a set of transactions which, if executed, would remove the inconsistency and retain the consistency of the knowledge base. To deal with various interactive systems, e.g. dialogues and dialogue planning, Gallagher and Guessoum (1992) proposed the method of 'anomaly removal' where the incremental removal of anomalous states models the interaction.

The knowledge-base update procedure can also be used in implementing abductive reasoning: inconsistent states are eliminated by forming hypotheses that would count as explanations for the input. PLUS explored abductive reasoning in interpretation, following the lines of Hobbs et al. (1990). Some results can be found in Underwood and Black (1994). The problem of selecting an appropriate transaction among different alternatives was also tackled with the help of conceptual filters. Work on this is reported in Meyer and Guessoum (1994).

2.3 Dialogue corpus

To provide relevant information for the design of the system, as well as for the test and evaluation tasks, a corpus of English, French and Swedish dialogues was collected in

the PLUS project. In this section we briefly discuss these empirical studies which also serve as an empirical basis for the claims made later in this thesis concerning robust dialogue management (see Section 5.1). A modified version of this section is Jokinen (1991), also included in the PLUS deliverable Nivre (1992a) as a report of the corpus collection work at UMIST.

Analysis of some of the PLUS dialogues in terms of CAA are described in Allwood and Haglund (1991), also included in Nivre (1992a).

2.3.1 General set-up for corpus collection

The Wizard-of-Oz method was used to get dialogues dealing with information seeking from the Yellow Pages. The method refers to a set-up where the users believe that they are communicating with a prototype dialogue system, while in reality there is another person in another room answering the questions and trying to simulate the future system. It has been extensively used in HCI studies, see e.g. Dahlbäck et al. (1993), Fraser and Gilbert (1991), Diaper (1986, 1989), Allen (1990).

The dialogues were collected by using a computer communication system prototype written by Afsaneh Haddadi at UMIST. This software allows human-human communication between two different Sun workstations, or between a Sun and a dumb terminal or terminal emulator. The software enables the input from the user to be shown on the wizard's screen as it is being typed, thus giving the wizard time to plan responses. The responses by the wizard, on the other hand, are buffered, so that the user cannot see the process of message composition. This gives the user the illusion of a computer being at the other end. The partners cannot interrupt each other, i.e. after sending the message by hitting the ESC-key, the participant has to wait until the other has responded before being given back the turn. The dialogues are recorded in a file which contains both user and wizard turns and is headed by the start time. The file also contains time information about the length of each turn, and records the user's mistypings and corrections.

The restrictions put on the wizard's side in the WOZ-studies were general. The wizard was a native speaker of English, and she was asked to be as helpful as possible given the limitations of a system that can only provide information from the Yellow Pages database. A set of canned responses was produced following initial trials, and these were made available in a window, but the wizard could also freely phrase her replies if the ready-made replies were unsuitable in the context. In the latter case, the wizard was asked to use language that was natural yet not too human-like, in order to maintain the illusion of computer responses.

The wizard also had a small database that contained appropriate information about car hire firms, restaurants and insurance companies needed in the dialogues. The database was compiled on the basis of the Greater Manchester Yellow Pages, concentrating on information in areas that were relevant for the scenarios. As the Yellow Pages does not divide restaurants into type categories, a local restaurant guide was used to help in this categorisation.

The aim was to use a wide range of human subjects, and especially subjects who were would-be users of the future system, to obtain a representative corpus. Half of the 35 subjects comprised UMIST administrative staff, bank clerks, radio presenters, electrical engineers, and a nurse. The other half consisted of 1st and 2nd year students and postgraduates, who were easily available in the academic environment. The age range of the subjects was 20–59.

The set-up of the dialogue session was as follows: subjects were given the instruction sheet that contained general information about the experiment and how to use the system, as well as the scenario sheet, and were then left to work on their own. An instructor was available to assist, if problems arose, but did not otherwise intervene in the conduct of the experiment. After the experimental session, the subjects were asked to fill in a questionnaire which contained questions about their prior experience with computers and their impression of the experiment.

2.3.2 Nature and type of dialogues

46 dialogues were collected on the following scenarios: hiring a car for a one-way trip (17), finding a restaurant that serves hot, spicy food (17), and finding an insurance office (12).² In this section, we present the collected dialogues from the point of view of their relevance to the purposes of this thesis.

How to begin and how to quit

As shown in a study by M. A. Richards and K. M. Underwood³, quoted by Fraser and Gilbert (1991), the system's responses have a big impact on the subject's queries. The manner in which conversations are opened is especially influential: explicit, impolite openings where the system gives precise information about the service and asks for a user request by imperative sentences elicit the most concise requests, while verbose but polite openings most commonly elicit polite responses.

It is not specified what factors Richards and Underwood used in determining politeness of the wizard's turn and the user's responses, but presumably conventionalised forms like 'I would like to', 'could you', 'please' were under consideration. However, our dialogues show that these phrases are used as conventionalised request forms rather than indicators of politeness. The subjects relied on the conventional relation between these phrases and requesting in general (as in "car hire firms please"), rather than 'softening' the request to be polite with the computer. Moreover, the wizard used fairly impolite language in terms of conventionalised politeness phrases which should have invoked less 'polite' behaviour in the subjects. Our dialogues also seemed to depend more on the subject's imagination than the system responses, and some of the subjects apparently chose to understand the experiment as a situation where they could try out the system's factual database limits rather than its natural language capability.

²For comparative purposes, 32 human-human terminal dialogues were also collected and analysed, but these are not reported in this study.

³'How should people and computers speak to each other?' *Interact* '84, pp 33-36, 1984

The software was designed so that the user had the first initiative in dialogues. This was generally considered puzzling, and the subjects kept asking the instructor “what shall I do”, “what kind of questions can I ask”, “can I ask ‘where are good restaurants in Manchester’”. Although the task was explained to the participants and they also had instruction sheets, it seems to be important that the first screen and front menu are informative enough: they should provide the user with information about the scope of the system and how to use it.

In general, the start was formulated in four different ways:⁴

1. Indirect conventionalised request

- (a) interrogative: *Can you tell me of any car hire firms in Entwistle?;*
Can you give me a list of indian restaurants as neas as possible to Levenshulme;
could you tell me where I can find a restaurant in central Manchester which serves indian or thai food;
could you please give me a selection of telephone numbers for Mexican and Indian Restaurants located in the centre of Manchester?
- (b) declarative: *I would like the number of a local insurance agent;*
i would like to find out about car hire services based around Bolton.

2. Indirect non-conventionalised request

- (a) interrogative, which can also be a direct question:
are there any restaurants near my home;
are there Indian restaurants in central Manchester?;
Do you know any Indian restaurants in Manchester?;
How many restaurants are there in Manchester that serve spicy food?
- (b) declarative, which can also be a statement:
I am looking for a restaurant;
I am looking for the names of Mexican restaurants in Chorlton cum Hardy;
I need a car;
I live in Entwistle outside Bolton and want to hire a car;
i need some addresses of insurance companies;
I want some insurance details.First of all, can I have some assessments for life insurance

3. Direct request

- (a) *please give me information on the indian restaurants in central Manchester;*
please supply details of small restaurants in the didsbury area;
Please tell me the phone numbers of car hire firms around Bolton

⁴Misprints are in the original contributions.

- (b) *indian restaurants in the centre of Manchester;*
car hire firms please

4. Direct question

What categories of restaurants do you list?

The three first types seem to be equally common, but only one direct question was found. This dealt with meta-knowledge: the user wanted to know the classification and coverage of the database. It should be noticed that indirect non-conventionalised requests (2) are fairly common and sometimes they are rather vague as well ('need a car', 'restaurants that serve spicy food'). Their interpretation as requests is based on the contextual knowledge of their occurrence in the beginning of the dialogue, as well as on world knowledge reasoning, filtered through the application model.

It is interesting that another conventional form of politeness, the closing ceremony, was rare in the WOZ-dialogues. Among the 30 people that participated in the human-computer dialogues, there were only two persons who spontaneously signalled the end of the dialogue by an explicit end marker like "that's all I need from you then, thanks for all your help!!!", "thank you; mission accomplished", "Thank you. Over and out.", "bye". The subjects seemed to use the system not as an equal dialogue partner, but as a machine, and they did not consider it necessary to go through the long closing ceremonies, or indicate that they have got all the information they wanted. When the task was accomplished, the conversation was considered finished, too.⁵ Also, only one quarter of the dialogues contained expressions of thanks at the end of the dialogue, although about two thirds (63%) of the subjects used *thank you* at some point in the dialogue.⁶

The lack of explicit closing is problematic for the system, since a long pause after a

⁵This was also the most common case when the instructor's help was needed: "I've got the information that I need, what shall I do next?"

⁶For the sake of interest, we can contrast this with the results obtained from human-human dialogues. Even though the set-up was the same, the subjects tended to be more polite and cooperative: 70% of the dialogues ended with thanking, and of these, 3/4 also contained explicit acceptance or indication of future plans in a form of the phrase: *Thank you, that is all I want to know.* and *Thanks, I'll give Hertz a ring.* All but one of the subjects thanked at the end.

completed task may mean either that the user is thinking what to ask next or has finished. Apparently, participants did not close their dialogues because they did not think closings were necessary when conversing with a computer: the task for which the dialogue served was completed when the requested list appeared on the screen. From the system's point of view, there seem to be three ways to infer that the conversation is over. First, after listing the required information, the system takes the initiative and, in the same turn, asks the user to make a decision about continuation: *Is this all you wanted to know?*, *Would you like to continue?* etc. The negative side of this alternative is that it automatically makes the dialogue awkward and non-human-like, especially if the question is always phrased in the same way and produced without any attention to the dialogue context. It may be appropriate after a fairly long dialogue, but appears clumsy after the very first user contribution if this is a direct question e.g. about the number of services in a particular location. The second alternative is to use mechanical time calculation: after a given waiting time, the system may decide that the conversation is over and close the connection. Finally, the third alternative is a combination of these two: after a certain waiting time, the system may take a more subtle initiative and ask if the user wants to continue the dialogue, and then, if no reply comes to this question, close the connection. This is used in some library catalogue systems.

User and wizard initiatives

The initiative taken by the wizard was restricted to specification and clarification questions. However, sometimes the wizard tried to simulate a dumb as opposed to an intelligent computer, and she produced questions and responses which were uncooperative in that they were too literally understood and not consistent with the overall requirements of the dialogue system. This was considered one of the main drawbacks of the dialogues collected: due to the inconsistency in the wizard's contributions, the users were forced to milk the requested information, and thus be engaged in frustrating dialogues. On the other hand, from the point of view of general dialogue studies such cases provide interesting material on human reasoning capabilities and resolution techniques

when dialogues are malfunctioning, expectations are unfulfilled and misunderstandings occur.

Most dialogues were more or less natural conversations, and subjects asked questions as if they were involved in real information seeking dialogues. However, their initiatives usually ended when the list of appropriate companies appeared on the screen, i.e. when they understood that the “mission was accomplished”. Sometimes the whole scenario was asked in the very first question (*Could you please give me a selection of telephone numbers for Mexican and Indian Restaurants located in the centre of Manchester.*), and the dialogue was over in two contributions. Students showed a more relaxed and creative attitude, and they frequently took initiatives in continuing the conversation with follow-up questions. For instance, in restaurant situations, questions like “what is the cheapest restaurant”, “which one is in the good food guide”, “which has car parking spaces”, “is it a walking distance from the bus station” were frequent questions. Some even produced conversation as if they were in a problem solving situation or naturally chatting with another person. Unfortunately, since the YP does not contain information beyond the names and addresses of the companies, the conversation often died out after several unsuccessful questions.

The subject’s evaluation of the dialogues

After the dialogues, each subject was given an evaluation form to write down their opinion of the system responses, their own performance and the dialogues in general. All the subjects accepted that they had been talking to a computer, and even if the wizard made obvious mistakes, the users accepted these as system errors that commonly occur when dealing with computers. The users seemed to be more concerned about quick and exact replies than fluent dialogues, because they did not expect computers to be fluent (they were usually surprised at the apparent capabilities of a computer to understand and produce such natural language).

Two middle-aged men kept typing one or two word queries, even though they were told to use ordinary language sentences. They were apparently fixed on the idea that they

were talking to a dumb computer which can only understand simple keywords, and this attitude is indicated in the evaluation form the subjects filled in after finishing the session. The first subject understood the whole situation as a test for his intellectual capabilities, and stated that his performance differed from face-to-face conversation “at the start, but as I got to understand how it worked it became easy to use”; the second one humbly wrote that his performance was “not very good. It is difficult to talk to a blank screen”.

Ethical aspects

We do not go into details of the ethical aspects of Wizard-of-Oz studies here. However, we want to point out that WOZ experiments as such pose moral and sometimes even legal questions about the set-up. It is of course important to tell people that they had actually been talking to a human who tried to act as a computer, otherwise wrong conceptions about the current state of NLP and AI could be spread. Usually the revelation of the true nature of the dialogue partner was met with a surprise and accepted as an interesting personal experience. However, there are potential ethical and juridical problems: although subjects come to the experiment freely, after announcement of a request for participants in a dialogue study, they are deliberately misled about the nature of the experiment.

2.3.3 Consequences for dialogue management

The collected WOZ-dialogues feature the same characteristics as noted by Dahlbäck and Jönsson (1989) and Beun and Bunt (1987) to be peculiar to information seeking dialogues and human-computer interaction in general. They are fairly straightforward question-answer sequences, and many of them lack conventionalised ‘politeness’-features like greetings, closings and thankings. They also show less indirect speech acts, and those encountered are usually conventionalised (*Can you tell me...*).

The lack of variety in user contributions seems to be a result of the simple scenarios

given to the subjects: the subjects did not see any problem in hiring a car and leaving it in another place. Moreover, the task itself, to seek information from the YP, was too straightforward, and the subjects were too familiar with it, to enable elaborate human-computer communication. The task can be understood as a systematic stepping through a well-defined path which consists of different alternative headings in the order defined by the YP organisation. The help given by the system is thus reduced to providing the alternatives quickly and clearly to the user, who could then just select the ones which suit her goals best.

From the point of view of dialogue management, it is, of course, very tempting to model the dialogues via a simple task structure, where communicative acts are tied to the system's need to get particular task parameters, cf. Grosz (1977), Bilange (1991). In fact, a finite state based dialogue manager was sketched in PLUS to show how simple methods can be used to capture the task-oriented aspects of the dialogues (Jokinen et al., 1991). However, the information in the YPs is huge, and it is not possible to predefine all the potential information seeking dialogues that will be made on the YP database this way. The system also needs to cope with several different "contextual implicatures" due to ellipsis, fragments and vagueness in the user requests. An overview of the problems as well as a way in which the knowledge base management systems can support the resolution of linguistic indeterminacy is given in the working paper Black and Jokinen (1991). Moreover, the aim in PLUS was to model dialogue management based on general communicative principles, independent from the task and domain information, and thus a more elaborated discourse model was designed (Jokinen et al., 1992).

2.4 Evaluation

PLUS had ambitious aims for the use of pragmatic and contextual reasoning in building a robust dialogue system which exhibits cooperative and flexible behaviour. The targets were achieved in embryonic prototypes of the modules (e.g. the Response Planner was considered a good illustration of how a PLUS system would plan its output

and how this is determined by the contextual situation), and in several experimental systems concerned with different relevant issues (reference resolution and disambiguation of word senses using abduction), but an integrated system could not be built due to the premature end of the project. Although the project contributed to better understanding of robust dialogue systems, the question remains whether the results are sufficiently mature to go into larger scale applications. The first system prototype is described in Bego et al. (1993).

The importance of PLUS for our work is that it set up a framework within which our research objectives can be formulated. It sketched a model of how pragmatic knowledge can be formalised and used in its context for limited scope domains of database queries. However, the ideas of a pragmatics based language understanding system were not realised in the form of “PLUS technology”, and more work is needed to spell out the power and advantages of such an approach. Our research aims to be a step in this direction: to specify more fully some of the principles of robust and cooperative dialogue management and their influence especially on the planning of system responses.

Chapter 3

Theoretical background

In this chapter we review the three main topics of the thesis: pragmatics, dialogue management and natural language generation. In section 1 we attempt to answer the question: what is pragmatics in natural language processing (NLP)? We investigate three issues we consider particularly important for pragmatics-based NLP: cooperation, coherence and robustness. In Section 2 we compare two different approaches to dialogue management: the intention-based and structural approaches, and assess their suitability for robust dialogue management. In Section 3 we review the state of the art in natural language generation, with respect to the production of cooperative system responses.

3.1 Pragmatics and natural language processing

Ahrenberg et al. (1994) list four requirements that are widely recognised as important for Natural Language Interfaces (NLI): ‘habitability’ (the user should be able to express commands and requests conveniently without transgressing the linguistic capabilities of the interface), ‘efficiency’ (the NLI should not slow down the interaction with the background system noticeably), ‘robustness’ (the system should be able to react sensibly to all input), and ‘transparency’ (the system’s capabilities and limitations should be evident to the user from experience). They argue in favour of customizing

the interaction with the help of Wizard-of-Oz studies (see Chapter 2.3 for details of this method), and are able to summarise the design of a dialogue system in two principles: ‘Asymmetry Principle’ (if there is a choice, prefer solutions that make the user learn from system contributions to solutions that require the system to learn from the user’s contributions) and ‘Sublanguage Principle’ (restrict the linguistic and general knowledge of the system to that which is needed to support the users’ tasks). As a conclusion, they advocate the following slogan as a design principle for simple service systems:

Prefer global pragmatics at design time to local pragmatics at run-time.

We concentrate on the first part of the slogan in this thesis. We search for a suitable definition of pragmatics and end up with a claim that the above mentioned requirements can be derived from a pragmatic approach that is related to communication in general.

3.1.1 What is pragmatics?

Pragmatics, as the term is used nowadays, goes back to the philosopher Charles Morris who sought to clarify the terms syntax, semantics and pragmatics (Morris, 1938).¹ According to him, syntax was concerned with the formal relations of signs to each other, semantics with the relation of signs to the objects to which the signs are applicable, and pragmatics with the relation of signs to their interpreters. As pointed out by Levinson (1983), there is no satisfactory definition of pragmatics which would neatly subsume all the different issues that pragmaticians have called pragmatic. Rather, pragmatics is best described through the issues and problems with which pragmaticians are concerned.

In the linguistic-philosophical tradition pragmatics refers to the ‘extra meaning’ of an utterance that remains outside truth-conditional semantics. This is summarised in the formula given by Gazdar (1979):

¹The distinction had already been made by C. S. Peirce who defined the three distinct areas within the field of semiotics, but the distinction has been made more generally known by Morris.

This definition calls for the definition of meaning, since the meaning that is left when truth conditions are subtracted may be narrower or broader depending on what phenomena are subsumed under the semantic theory. Levinson (1983) gives seven putative meaning components or inferential relations of an utterance, which we list in Figure 3.1. If semantics is understood in its narrowest, truth-conditional form, then ‘extra meaning’ can be said to comprise the other meaning components, i.e. the scope of pragmatics concerns the items (2) - (7).

- | |
|--|
| <ol style="list-style-type: none"> 1. truth-conditions or entailments 2. conventional implicatures 3. presuppositions 4. felicity conditions 5. generalized conversational implicature 6. particularized conversational implicature 7. inferences based on conversational structure |
|--|

Figure 3.1: Elements of the communicational content of an utterance. Taken from Levinson (1983, p. 14).

In NLP, pragmatics has been connected with the aims of AI research in general: to model the understanding of the world. Since natural language provides a means to talk about and understand more about the world, the models can be tested and improved with the help of natural language processing. Pragmatic issues include the items (2) – (7), but their study concerns the interaction between linguistic knowledge and world knowledge, rather than the phenomena as such. The emphasis is on the inferential nature of pragmatic phenomena: to understand a sentence, not to mention texts and dialogues, the relationship between linguistic phrases and their contexts must be identified. This is done by making inferences that connect what is said to what is mutually assumed or said before.

However, there are two problems with this notion of pragmatics, also pointed out by Levinson. First, a huge amount of contextual knowledge is required, and second, the type and nature of contextual knowledge is difficult to determine. Contextual knowl-

edge ranges from linguistic issues such as reference resolution and topic information to world knowledge such as causal and temporal chains, task structure and planning sequences, and we are faced with an ever-expanding context. The characterisation of the context usually starts by drawing a distinction between the *actual* situation in which the utterance occurs, consisting of all the possible contextual features, and the *abstract* situation, which contains only those features which are relevant (culturally, linguistically or for the purposes of a working system) for the interpretation of the utterance. However, it is difficult to provide a complete list of contextual features, and in practical applications, only a few are taken into account.

Much work has been done to characterise different aspects of language use. In the Systemic Grammar framework (Halliday, 1973) these aspects are divided into ‘field’ (the subject matter), ‘tenor’ (interlocutors’ interpersonal roles and relations) and ‘mode’ (the form and situation of the communication). On these lines, Hovy (1988a, p. 17) gives the following high-level categorisation of pragmatic information needed in NLP applications:

- interlocutors’ personal characteristics: factual knowledge, opinions, emotional states, interpersonal relationship (hearer’s emotions toward speaker, depth of acquaintance, relative social status), etc.
- the speaker’s goals with respect to the hearer: effects on future behaviour, opinions, relative status, etc.
- conversational atmosphere: tone, time, physical setting etc.

These aspects can be compared to the view which sees pragmatics as related to communication in general. In its widest sense, pragmatics is understood as embracing all factors which influence communication between human beings, but it is useful to narrow down the scope by concentrating on the “phenomena that communicators consciously ...have to take into account” (Allwood, 1976). The above-mentioned categorisation of pragmatic information can thus be accounted for by referring to the general requirements for communication (e.g. to be in contact and to be able to perceive and understand communicative contributions), and to the normative social obligations that communication creates between the communicators (to evaluate information and report

the result of the evaluation, consider the other communicator as a rational, motivated and competent agent).²

While reference to general communicative principles may seem abstract from the point of view of practical NLP applications, we believe that the functionality of such systems can be significantly improved only if the design decisions can rely on an adequate theoretical basis. A lot of research has been conducted with the aim of developing dialogue systems that would not only provide correct responses, but would also show cooperative and robust behaviour in general. However, while acknowledging the abundance of insightful research, we think that one of the most important areas of neglect in existing dialogue systems (such as HAM-ANS, COOP, SUNDIAL) is that they lack a general theory of what it means to behave cooperatively.³ We believe that it is vital to use pragmatics extensively to improve the system's user-friendliness and generality, and especially, that the notion of pragmatics in dialogue systems should be about the system's communicative capability.

Below we will discuss three issues which we regard as 'pragmatic' in the sense that they are related to communicative capability, and which we consider important in the design and development of flexible, user-friendly natural language dialogue systems.

1. **Cooperativeness:** informative and helpful responses given by the system, based on the evaluation of the user's contribution in the dialogue context, and conforming to the communicative principles of 'ideal cooperation'.
2. **Coherence:** smooth information flow in the dialogue in which the dialogue parts 'hang together'.
3. **Robustness:** the system's communicative capability, especially its external user-friendliness and the precision level on which vague or misunderstood input is clarified.

²A more detailed discussion of the levels of communication can be found in Chapter 4.

³To be fair, these system did not aim at pragmatics-based NLP, but rather, at showing how the appropriateness of system responses can be improved by paying attention to some particular aspects of dialogue context.

3.1.2 Cooperation

Cooperativeness is commonly understood as one of the basic requirements of successful communication. However, it is difficult to explain what are the characteristics of cooperative behaviour and how cooperativeness works in actual situations.

Usually it has been regarded as a realisation of the Cooperative Principle of Grice (1975):

Make your conversational contribution such as required, at the stage at which it occurs, by the accepted purpose or direction of the talk exchange in which you are engaged.

and especially, of the four conversational maxims that characterise the Cooperative Principle:

- *Quantity*: be informative, but do not make your contributions more or less informative than is required,
- *Quality*: be truthful, do not say anything for which you lack adequate evidence,
- *Relation*: be relevant,
- *Manner*: be brief and orderly, avoid obscurity and ambiguity.

The Gricean approach has been criticised both from the philosophical-linguistic and the NLP view point. The philosophical-linguistic arguments deal with the fact that people often break the maxims, but do not appear to be uncooperative, or they observe the maxims and still cause misunderstandings. Actually, Grice himself discussed the ways in which a participant may fail to fulfil the maxims, and considered flouting of the maxims a situation which characteristically gives rise to a conversational implicature (e.g. the use of metaphor and irony.) However, maxims are intended to serve the purpose of “maximally effective exchange of information”, and thus they can be successfully applied only to dialogues which deal with factual information. Although Grice considers conversations as joint enterprises, the maxims also tacitly assume that communication is a rigid exchange of information where each utterance fulfils all the

requirements exhaustively and at once. Since dialogues, including factual information seeking dialogues, are best characterised as negotiations rather than straightforward question-answer sequences (Pollack et al., 1982; Roulet, 1986; Moeschler, 1989; Moore & Swartout, 1990; Clark & Wilkes-Gibbs, 1990), cooperativeness should be defined with respect to both participants, who, pushing their own goals and at the same time showing consideration of the partner's goals, collaborate in a joint task to achieve a mutually acceptable result.

Cooperation can also be looked at from a wider perspective and considered a sign of the communicators' communicative competence. Cooperativeness is not a means to guarantee comprehensibility of individual utterances or effective communication, but rather an effect that emerges from the dialogue partners' willingness to continue the dialogue and their ability to make relevant contributions until both have achieved their goals and are satisfied with the result.

In his thesis, Allwood (1976) studied motivated rational activity as a basis of linguistic communication. He sees cooperation as a type of interaction between normal rational agents who are mutually considering each other when trying to achieve one or more common purposes. Communication is a fundamentally cooperative activity which creates social obligations between the agents, and would break down if a serious conflict made cooperation impossible.⁴

Allwood criticises Grice because, despite his intentions, Grice does not pay enough attention to the fundamentally cooperative nature of conversation. For instance, Grice's maxims do not take the speaker's consideration of the receiver into account. Instead, Allwood derives Gricean maxims as special cases from the ideal cooperation and social obligations that communicators commit themselves to, which in turn can be derived from ethics, rationality and agency. For instance, the maxim of Quality is a direct consequence of the social obligation to take responsibility for the expressed beliefs and attitudes: to tell lies is to commit oneself to something which one may not be able to maintain. Similarly, the maxim of Manner gets its different forms from the different

⁴For details of Allwood's theory, see Chapter 4.

norms which constrain adequate reporting of the information.

On the other hand, cooperation should not be understood as total compliance with the social norms and obligations. Acknowledging Grice (1975), Allwood (1976) also discusses cooperative understanding which arises from the clash between norms and obligations of rational agenthood, or from the flouting of the norms. In these cases, constructive reinterpretation of the contribution takes place to retain cooperativeness. The speaker trusts the partner to act as a rational agent who trusts *her* partner, i.e. the speaker, to act as a rational agent, and thus content which is blatantly irrelevant or incorrect is reinterpreted according to the constraints of ideal cooperation, mutual purpose and ethical consideration.

Furthermore, as pointed out by Galliers (1989, p. 199), situations where the agents have conflicting goals are common in the real world, and actually play “a crucial and positive role in the maintenance and evolution of cooperation in social systems”. Cooperation is thus not only benevolent conforming to social norms and obligations, but active seeking for the achievement of goals and the resolution of conflict if the agents have conflicting goals. Allwood (1976) does not discuss the management of communication when the participants lack a common purpose or they have conflicting goals, but this can be subsumed under what is called ‘Interactive Communication Management’ in Allwood and Haglund (1991), or more globally, under the evaluation process of the partner’s goals and intentions with respect to one’s own. An important departure point in conflict resolution is the partners’ willingness to comply with the principles of ‘ideal cooperation’: as long as the partner is treated as a rational agent whose activity is purposeful and motivated, communication is possible to resolve conflicts.

As for uncooperativeness, we must distinguish the situation where the partner intends to cooperate but lacks the ability to do so, from the situation where uncooperativeness is intentional and comprises various degrees of lying and deception. In the former case the failure is unintentional and described as misunderstanding rather than uncooperativeness, while in the latter case we have genuine uncooperativeness. Of course, intentional uncooperativeness can be undetected (the partner is lying, but the agent does not notice this), but in this case, the partner’s behaviour is considered cooperative

as long as the agent finds it consistent and helpful.

3.1.3 Coherence

In some intuitive sense, coherence means that the discourse (text or dialogue) talks about the same theme. On the surface level, individual sentences and dialogue contributions may concern unrelated objects and events, but as a whole, the discourse parts can be linked together into a coherent communicative whole. In general, there appear to be two types of links: those that hold between ideas or propositions, and those that hold between discourse objects or between objects and events. The links between ideas have been modelled by general domain independent relations, while the links between objects and events have been captured with the help of concepts such as topic and focus, or salient discourse referents.

In text linguistics and text generation, much work has been done on the conditions and requirements for coherent texts (Halliday & Hasan, 1976; de Beaugrande & Dressler, 1981; Hovy, 1986, 1988b, 1990, 1991; Hovy & McCoy, 1989; Sibun, 1991), especially on how to organise the content into a coherent paragraph. The work has naturally centred on the links between successive discourse segments, being referred to as conjunctive relations (Halliday & Hasan, 1976), coherence relations such as Elaboration, Parallel and Contrast (Hobbs, 1979) or rhetorical relations such as Motivation, Evidence, and Sequence (Mann & Thompson, 1987).

The problem with these kinds of domain-independent coherence relations is that they are actually dependent on domain knowledge and on the particular domain relations that the domain supports. Kittredge et al. (1991) argue that ‘Domain Communication Knowledge’ is important and should be explicitly represented in text planning. They used Mann and Thompson’s (1987) rhetorical relations to generate weather forecasts, and noticed that the domain not only restricts the type of relations needed in the task, but also the appropriateness of the relations to express domain information. For instance, the Sequence-relation requires knowledge of what domain facts are appropriate to include in the sequencing relation.

Coherence is also dependent on the speaker's ability to find a relevant causal or motivating relation between the contributions. For instance, the discourse (2):

(2) *John took a train from Paris to Istanbul. He likes spinach.*

which is an example of an incoherent discourse in Hobbs (1979), appears coherent, if it is known, or can be inferred, that Istanbul is a good place to buy spinach. The second sentence then provides explanation or motivation of John's taking a train to Istanbul. Hobbs acknowledges that coherence relations are not absolute relations, but that they hold between sentential units to a greater or lesser degree, depending on how salient the axioms were that were used to establish the relation. However, the salience of the axioms is not constant across all speakers, and not necessarily constant in the same discourse, but depends on the speakers' knowledge about what counts as a possible domain relation. Consequently, coherence relations are tied to user modelling: what the speaker knows and believes that the partner knows about the domain. The same discourse may appear more coherent to some speakers than it does to other speakers, because their background knowledge is different.

In dialogues, coherence is a more complicated issue than in monologue texts, because coherence is not controlled by one speaker, but built together by the participants in a joint communicative activity. It serves the participants in their attempts to achieve their own goals and understand the partner's goals in the most efficient way, and is thus associated with dialogue management in general, besides the domain rules for producing thematically coherent responses. For instance, the recent work on interactive explanation generation (Moore & Pollack, 1992; Moore & Paris, 1989) shows that knowledge of structural rhetorical relations is not enough to enable an expert system to respond appropriately to the user's follow-up questions (and thus to guarantee that the dialogue is coherent). This must be combined with knowledge about the user's intentions and goals, since the same rhetorical relation can be used for different communicative purposes and the same communicative goal can be achieved via different rhetorical relations.

Different approaches to dialogue management, which will be discussed in detail in Sec-

tion 3.2, use different means to deal with the coherence of dialogues: the structural approach relies on explicit and implicit structures in the dialogue (Sinclair & Coulthard, 1975; Grosz, 1977; Bilange, 1992), while the intention-based approach deals with the intentions of the speakers and the coherence of their plans (Allen & Perrault, 1980; Cohen & Perrault, 1979; Cohen & Levesque, 1985; McKeivitt et al., 1992). The linking of ideas and propositions into a coherent discourse is based on the respective assumptions that coherence is determined by the speaker’s conforming to a predefined dialogue structure, or by her recognising the partner’s plans and fulfilling consistent subgoals.

The primed context and the links between discourse referents have been modelled with the help of the notion ‘focus’ (e.g. Sidner, 1979; McKeown, 1985; Grosz, Joshi, & Weinstein, 1983; Grosz & Sidner, 1986; McCoy & Cheng, 1990). The focussed entity is algorithmically identified from the set of discourse referents mainly on syntactic and semantic criteria (Sidner (1979) provides extensive work on pronominalisation of focussed discourse referents), but also the role of context has been emphasised: Pignataro (1988) attempts to define ‘topic’ and ‘focus’ in relation to context, and Reichman (1985) differentiates discourse referents along a four-level focus continuum associated with her ‘context spaces’ which structure the discourse.

Although the main function of ‘focus’ has been to restrict the search space for relevant information, determine referring expressions, and select surface level expressions (i.e. it deals with the surface level ties or with ‘cohesion’ rather than coherence, (Halliday & Hasan, 1976)), the tacit assumption is that the discourse is coherent, if it talks about the same set of entities. For instance, Grosz et al. (1983) use the notions of ‘global focus’ and ‘centering’ to distinguish between *global* and *local coherence*, respectively.⁵ ‘Global coherence’ refers to the ways in which larger segments of discourse relate to each other, and accordingly, global focus refers to a set of entities that are relevant to the overall discourse. ‘Local coherence’ deals with individual sentences and their combination into larger discourse segments, and accordingly, centering refers to a more local focussing process which identifies a single entity as the most central one in an individual sentence. We come back to this distinction in Chapter 7 when we discuss

⁵The distinction was originally drawn by Grosz (1977), who talks about *global* and *immediate focus*.

the thematic coherence of dialogues.

Hobbs (1979) uses example (2) (p. 36) to argue that coherence does not result from the discourse being about the same entity or the same set of entities, but from the speakers' need to be understood. This need drives the speakers to seek for an appropriate coherence relation, and the fact that a discourse is of some set of entities is a "trace of the deeper processes of coherence".⁶ We agree with Hobbs that coherence is due to the communicators' desire to make sense of the discourse, but we would also make a stronger claim, namely that the communicators are *obliged* to produce coherent discourse, given that they act as rational, cooperative agents. Hence, coherence is motivated not only by the speakers' need to be understood, but also by their compliance with the principles of ideal cooperation and rationality (cf. Allwood, 1992).

On the other hand, the amount of reasoning that the speakers need to do in order to determine whether the discourse is coherent or not, depends on the primed context in which the analysis takes place. For instance, the discourse (2) is odd in isolation, but it is easy to direct the reader towards the intended interpretation, if the context is extended e.g. by adding a title like *Turkey - the world's best spinach provider*, and an extra clause like *And he was determined to find the best spinach in the world* at the end. An analogous point was made by McCoy and Cheng (1990) with respect to different focussing effects found in the discourse. They noticed that contrary to what Garrod and Sanford (1983) claim, the discourse

- (3) Harry fell several times.
 ? *The snow* was cold and wet.

seems reasonable and the reference to 'snow' is both relevant and acceptable, if the sentences are embedded in a longer context which includes e.g. the title "Learning to ski". The explicit reference to skiing causes the interpretation to be oriented towards the domain concept 'snow', which thus contributes to the coherence of the discourse.

⁶Hobbs argues in favour of abductive reasoning which can be used to 'explain' the observed sentence in terms of semantic predicates, and by instantiating the predicates appropriately, coherence appears as a side-effect.

We regard the set of entities that the discourse is about as an important tool in the analysis and evaluation of the contributions, since it provides the common ground or the primed context within which communication takes place. In generation in particular, this set provides the context within which thematic coherence is determined: topic shifts are based on the knowledge about how the discourse can continue cooperatively, given the Central Concept of the contribution and the context of what has been talked about.

3.1.4 Robustness

Robustness can be looked at from two different view-points: the application oriented system building view-point with the aim of producing good user interfaces, and the more theoretical system design view-point with the aim of generality of solutions. These are not necessarily contradictory, since the aim in both is to build helpful and cooperative dialogue systems that are capable of producing efficient, accurate and appropriate responses. Rather, their difference lies in the use of linguistic and pragmatic knowledge in dialogue management, especially in the way the system can reason about different continuation possibilities.

Hayes and Reddy (1983) define robustness as the system's ability to be able to react sensibly to all input. However, wide coverage as such is not interesting, since, as pointed out in PLUS (Cavalli et al., 1992a), this could be trivialised to quasi-conversational robustness by responding "Please re-phrase" to all problematic requests. Moreover, coverage can also be seen as closely tied to the particular application: the system can be tailored according to the specific needs peculiar to that domain. This has been amply debated in terms of domain dependency and the sublanguage approach, see e.g. Grishman and Kittredge (1986), Kittredge et al. (1991), Tsujii et al. (1992), Ahrenberg et al. (1994).

We claim that a dialogue system's desired behavior should be grounded on pragmatic principles which define cooperative and appropriate communication in general. Although the mouse and menu approach seems, at the moment at least, to provide a more

robust technology to human-computer interaction than natural language approach, we want to emphasise that robustness should not be restricted to interface issues only, but the quality of the system responses should be taken into account as well. This is an issue that requires pragmatic knowledge about the agents communicative competence. In the fast-developing multimodal environments, ‘robustness’ could thus be associated with the external enablements of communication like a user-friendly interface as well as with the adequacy and precision with which responses are planned.

We study robustness issues in the light of empirical dialogue studies in Section 5.1. We come up with four requirements for robust and cooperative dialogue management, related to those given by Ahrenberg et al. (1994) (see above p. 27), and we show how these requirements are based on the pragmatic principles of rational cooperative communication.

Robustness in the sense of communicative competence of the system was also one of the aims in PLUS. To give an overview of the full scale of the aims for this kind of robustness, Figure 3.2 lists features that were discussed in PLUS as desiderata for a robust, conversationally adequate system (Allwood et al., 1991).

3.2 Approaches to dialogue management

We can distinguish two different approaches to dialogue management. Both of them have a common basis in regarding natural language as purposeful behaviour, but they differ in general theoretical orientation about how this behaviour is to be described. This is due to the different scientific environment in which the approaches have been developed. We call these approaches the *Intention-based approach* and the *Structural approach*. We also briefly discuss *Conversation Analysis* whose insights on the study of naturally occurring dialogues have had impact on the terminology of human-computer interaction.

Users:	A variety of different kinds of users in different situations and activities asking about several different Yellow Pages topics.
Contact:	Long pauses and unfinished dialogues.
Perception:	Bad connections, spelling errors, users mistakenly hitting the keys (because of fatigue, clumsiness, drunkenness, etc.).
Understanding:	Ungrammatical input, terminological mistakes, mistakes in thinking, logic, unknown terms or statements, statements or presuppositions which the system believes to be false.
Attitudes:	User aggression, anger, swearing, etc., user misuse, sabotage, user inability to answer system's questions, user's demand that system evaluate something.
Processing:	<p>The system should be able to handle as many of the input difficulties as possible while giving as qualitatively good output as possible.</p> <p>...</p> <p>Qualitative processing means that the system should reach high levels of understanding, as quickly as possible, even when perception is inadequate or relevant background information is lacking.</p>
Output:	<p>The system should choose replies which pay attention to relevance hierarchies (cf. Allwood (1984) and Gricean type maxims of ethics and rationality.</p> <p>...</p> <p>The general problem of whether the system should go for quality or quantity has to be faced. Should we be able to do well always or should we accept less quality in order to handle more different users in more different activities with more different topics confronting the system with more types of 'corrupt input', etc?</p>

Figure 3.2: Robustness features in PLUS. Taken from Allwood et al. (1991)

3.2.1 The Intention-based approach

This approach has a logic-philosophical background and is rooted in the work on speech acts by Austin (1962) and Searle (1969). Language is viewed as action: utterances serve not only to express propositions, but also to perform actions. Communication is thus described in terms of beliefs and intentions of the dialogue participants, and successful communication means that the hearer recognises the speaker's purpose in using language. Utterances may contain performative verbs that explicitly indicate the act performed, but in general, utterances are implicit performatives, and the act is to be conventionally inferred on the basis of the utterance context.

There are different variations of this approach, mainly emphasising different aspects of the communication. For instance, in Dynamic Interpretation Theory (Bunt, 1990), implemented in the TENDUM project (Bunt et al., 1984), the effect of dialogue acts is seen via their context-changing potential, while in CAA (Allwood, 1976; Allwood & Haglund, 1991) communication is understood as social activity between rational agents and dialogue contributions express and evoke normative obligations about the appropriate continuations.

Speech Act Theory and its development

Austin (1962) characterised speech acts by three different acts being simultaneously performed when a sentence is uttered: locutionary act (the mere utterance of a sentence with determinate sense and reference), illocutionary act (performance of a certain act by virtue of the conventional force associated with it), and perlocutionary act (the result achieved by performing the act, bringing about of effects on the listener). The illocutionary act is associated with an illocutionary force that constitutes the type of act performed. Illocutionary acts are considered conventional, whereas perlocutionary acts depend on the particular situation in which the utterance is uttered. An utterance which carries particular illocutionary force is called 'performative' and verbs like *promise*, *bet*, *apologise*, *declare* are 'performative verbs'. In later developments of the speech act theory, performatives were divided into explicit and implicit, depending on

whether the utterance contained an overt performative verb or whether such a performative prefix should be postulated in the semantic ‘deep structure’ of the utterance.

Searle (1969) developed Austin’s theory by drawing attention to the felicity conditions that need to be fulfilled if the act is to be carried out successfully. An (explicit or implicit) performative cannot be true or false, but it can go wrong, be infelicitous. Searle suggested that there are four kinds of felicity conditions: propositional content, preparatory, sincerity, and essential conditions. They deal with the restrictions on the content of the sentence uttered, on the real-world prerequisites of each illocutionary act, on the speaker’s beliefs and intentions for each kind of action, and on the constitutive features of an action (‘doing X counts as Y’), respectively.

A problem of speech act theory is how the speaker’s mental state can be linked to utterances, i.e. how the speaker’s intentions and beliefs can be assigned to the recognition of speech act types. In the early work, this was done in two ways: utterances were classified according to the illocutionary acts performed by uttering them (cf. Austin’s (1962) attempt at a taxonomy of performative verbs), or by felicity conditions that need to be met if the act is to be felicitous (cf. Searle’s (1969) comparison between requests and warnings). Both ways suffer from the fact that no simple correspondence between utterance form and utterance force exists. Actual language use is diverse, and a straightforward mapping of speech act force to sentences in their context is difficult. Moreover, the interpretation of indirect speech acts poses a major difficulty: they are related to non-conventional conversational implicatures which require contextual reasoning beyond the literal force of an utterance.

Gazdar (1979) summarises the relation between illocutionary force and the sentence form in two rules: (1) explicit performatives have the force named by the performative verb, (2) otherwise the three main sentence types, namely imperative, declarative and interrogative, are associated with conventional illocutionary forces of ordering or requesting, stating and questioning, respectively. Levinson (1983, p. 263) calls this view *literal force hypothesis*, and he also points out that it renders most language use indirect: all speech acts other than those corresponding to the conventional literal force of the sentence type have an additional, inferred force, and must therefore be

considered indirect. For instance, the several ways of requesting are seldom expressed in imperative form, and the indirect force must be somehow inferred from the surface form of an utterance. Levinson (1983) proposes the ‘context-change theory of speech acts’ whereby the speech acts are understood as operations on the context, i.e. as functions from contexts to contexts. When a sentence is uttered, not only its meaning is expressed, but the set of background assumptions is changed as well. This view is also advocated by Bunt (1990, 1991) in his theory of dynamic interpretation (see below).

Speech acts can also be based on a more general theory of communication and be regarded as a special class of communicative acts (Allwood, 1976). ‘Illocutionary force’ can be associated with the communicative intention of the speaker, and felicity conditions be predictable from general considerations of co-operativeness and rationality of the locutors. In that way, utterances are understood as means to carry the speaker’s beliefs and intentions (that describe the speaker’s mental state), and thus they can change the mental states of the interlocutors. A similar view is also advocated by Cohen and Levesque (1990b) who consider specific utterance events in the context of the speaker’s and hearer’s mental states, and derive the different effects of the acts from general principles of rational agenthood and cooperative interaction. The illocutionary act types can be used as convenient labels of the speaker’s mental state, but they do not constitute an explanation of a dialogue. It should be noticed, however, that even though illocutionary act recognition may be unnecessary, it can be practically useful (Appelt, 1985).

Dynamic Interpretation Theory

Here we briefly review the main characteristics of dialogue modelling based on the beliefs and intentions of the speakers. The discussion is based on Beun (1989), which is in line with the ideas of dynamic interpretation as developed by Bunt (1990, 1991), which underlie the dialogue modelling of TENDUM (Bunt et al., 1984). Beun studies the interpretation of ‘declarative questions’ like “And that’s on Saturday too?” in information dialogues, and sketches a framework to identify their communicative function (either an answer or a verification question) on the basis of linguistic features of the

utterance and contextual evidence.

Based on Grice's (1957) definition of the meaning-nn (non-natural meaning), Beun (1989) considers communication a complex kind of intention that is satisfied just by being recognised: the speaker intends to cause some effect in the hearer, and this effect is intended to be achieved by the hearer's recognising this intention. In this process, the speaker's communicative intention becomes mutual knowledge to the speaker and to the receiver.

A communicative act is a function from context to context, where the context refers to the mental state of the participants (their beliefs, expectations, wants, intentions etc.) A communicative function is defined as a function from propositions to actions and denoted by a tuple called 'function structure', which contains utterance features, sentence type, pragmatic particles, and prosodics. A communicative function thus identifies communicative acts, on the basis of the observed utterance features, while the act itself is determined on the basis of context changes. Of course, the idea behind this is that the same communicative act can be performed by different communicative functions and different communicative functions can contribute to different acts. Linguistic and contextual features of utterances, such as the sentence type, intonation and certain pragmatic particles, were identified in empirical research on information dialogues, and they are directly linked to the felicity conditions for specific communicative acts.

The beliefs and intentions of an agent and the acts performed by the agent are represented in a logical language, and beliefs obey the standard logic axioms. Intentions are supposed to be consistent with the agent's belief, and they are related to each other via a set of axioms. Reasoning about communicative act consequences is performed on the basis of Perrault's (1987) default theory which enables reasoning without complete knowledge of the circumstances.

The context is updated by inferences based on each utterance and its previous context. Communication is succesful, if the felicity conditions are recognized by the recipient and the conditions become mutual belief. In other words, an utterance communicates

its felicity conditions. The felicity conditions are related to what a speaker intends the hearer to believe. Thus Beun speaks about a preferred set of felicity conditions as a mental state description rather than as a direct or indirect interpretation of a communicative act. The preferred set of felicity conditions associated with the utterance of a declarative sentence is that the speaker intends the hearer to believe something. The preferred set associated with the utterance of an interrogative is that the speaker intends the hearer to believe that the speaker wants to know something. If the context contains evidence that the preferred (default) set of felicity conditions is inconsistent with the hearer's knowledge of the context, the set is rejected and a less preferred set chosen as the interpretation of the utterance.

Discussion

General AI planning techniques have been used in computational applications to plan sequences of speech acts. Speech acts are modelled as planning operators with preconditions and effects, these being defined on the basis of the speaker's and hearer's beliefs and wants. The first system of this kind was the one described in Cohen and Perrault (1979), followed by further work by Allen and Perrault (1980), Allen (1983) and Appelt (1985). Most of the work within intention-based approaches has concentrated on single question-answer pairs, not whole dialogues with several contributions. The connection between intentions need not be only local, however, as e.g. the work by Carletta (1991) has shown: her intention-based dialogue games cover whole dialogues. Also McKevitt et al. (1992) show that coherence of discourse can be modelled as a sequence of intentions.

The problem with the intention-based approach is the recognition of speech acts: there is a diversity of actual language usage and no simple correspondence between utterance form and utterance force exists. On the one hand, a direct mapping from utterance features to the intentions of the speaker suffers from the fact that the conventional form of the utterance can be used in several different ways in different contexts, and thus the mapping algorithm requires extensive world knowledge about actions, causality, temporal issues, people's intentions in general, etc. Beun (1989, p. 125) for instance

notices this and states that “speech act theory mainly contributes to the idea that language use can be conceived as the performace of actions, and therefore should be considered part of a more general theory of planning and acting”. On the other hand, a plan-based approach, where the speaker’s intentions are embedded in the recognition of her plan, tends to tie dialogue structure to the task structure (Grosz, 1977; Appelt, 1985). Following Litman (1985), two planning levels (domain and dialogue planning) are separated, but the recognition of plans is considered more pertinent to dialogue management than the recognition of communicative strategies: e.g. Litman and Allen (1984) regard simple discourse conventions as sufficient, and concentrate on metaplans as a way of liberating domain planning from a strict task structure.

In the more recent work on speech act theory, Cohen and Levesque (1985, 1990a, 1990b) abandon speech act classification in terms of preconditions and effects, and specify the acts according to general principles of rational agenthood and cooperation. Given the agents’ mental states and the specific utterance event, an appropriate action can be generated and interpreted on the basis of the agents’ understanding of the general principles that govern the relation between mental states and speech action. Galliers (1989), who extended Cohen and Levesque’s framework to cover conflict situations, points out that problems with the rigid speech act recognition can be avoided by the shift of emphasis from the act itself to the speaker’s goal that the hearer should adopt a particular mental state. Our work on Constructive Dialogue Management is in accordance with the claims made by Cohen and Levesque, although we differ from them in our attempts to identify different communicative principles in dialogue management and their relative impact especially on the planning of responses.

3.2.2 The Structural approach

The structural approach assumes that dialogues can be analysed and generated according to some structural rules which govern possible dialogue units and constrain appropriate continuations. A dialogue grammar defines dialogue units such as moves, exchanges and segments, as well as their possible combinations. There is a wealth of

research within this approach, ranging from theoretical work by Sinclair and Coulthard (1975), Reichman (1985), Roulet (1986), Moeschler (1989), Suchman (1987), Scha and Polanyi (1988) to applications in dialogue systems by Grosz and Sidner (1986) and projects like LOQUI (Wachtel, 1986) and SUNDIAL (Bilange, 1992).

Dialogue Grammar is used to guide both analysis and generation of utterances. In LOQUI (Wachtel, 1986), dialogue rules are simple rewriting rules, where the arrow is to be read ‘consists of’:

$$(4) \quad \begin{aligned} & Dialogue \rightarrow Exchange + \\ & Exchange \rightarrow Init (Exch) Resp \end{aligned}$$

The first rule says that a dialogue consists of one or more exchanges which on their part consist of an initiative move, optional embedded exchange, and a response move. The grammar can handle simple question and answer sequences with optional subdialogues, but not three-turn exchanges or dialogues with missing response moves.

A more elaborated dialogue grammar is introduced in Bilange (1992), and used in SUNDIAL. This model is based on the work of the Geneva school of discourse studies, notably that of Roulet (1986) and Moeschler (1989). It consists of four hierarchially organised levels: ‘transactions’, ‘exchanges’, ‘interventions’ and ‘dialogue acts’. Dialogue acts are the basic structural elements, consisting of preconditions (informative or conversational goals, mental state conditions, and a dialogue situation) and effects (informative or conversational effects, new mental states and a new dialogue situation). Interventions are made up of one or more dialogue acts, and exchanges are made of interventions and/or exchanges. An exchange need not be a two-turn pair, three-turn exchanges are possible. Moreover, a response move need not be a single move but can be an exchange as well, thus allowing flexibility in the dialogue structure, since a whole exchange can be a response. Dialogue rules are used to plan possible dialogue continuations for both dialogue participants. In order to capture different aspects of dialogues, four different types of dialogue rules are used: ‘Grammar Rules’ (making predictions for the next turn according to the dialogue grammar), ‘Dialogue Control

Rules’ (recognising a failure situation in which to exercise control), ‘Conversational Rules’ (maintaining a smooth dialogue in terms of coherence and explicitness), and ‘Corrective Rules’ (bringing about necessary changes in the dialogue history due to the lack of expressive power in the effects of dialogue acts). The rules operate in the different phases of the generation cycle.

PLUS Dialogue Structure Rules

In PLUS, dialogue structure rules were also developed, although their status in the overall dialogue management approach was left unclear. The following is based on one of the project deliverables (Jokinen et al., 1992).

The Dialogue Structure Rules define units called ‘Move’, ‘Exchange’, ‘Development’ and ‘Interaction’, each augmented with a set of features which represent the information encoded by the unit. The information concerns the owner (i.e. the speaker), the communicative function (specification, clarification, confirmation) and the system task (information that is needed to fulfil the task for which the Exchange/Development was initiated, e.g. to know the location of a ‘ServiceSupplier’). The rules provide top-down predictions about the acceptable continuations of the dialogue. They are specified as follows:⁷

$$\begin{aligned}
 (5) \quad & I \rightarrow M_o E+ \\
 & E[f, x, s] \rightarrow M[i, s] (D[f1, x, p]) M[r, p] (D[f2, x, s]) (M[e, s] (D[f3, x, p])) \\
 & D[f, x, s] \rightarrow E[f, x2, s] (D[f1, x3, s])
 \end{aligned}$$

The first rule states that an Interaction between a user and the system consists of an initialising meta-act by the system (M_o : *Welcome to the electronic YP. How can I help you?*), and one or more Exchanges. The next rule captures the three-move structure of an Exchange: an Exchange consists of the initiative Move by a speaker, followed by the partner’s response Move, followed by an optional evaluative Move by the speaker. Each

⁷‘+’ means that the constituent must occur at least once, parentheses distinguish optional constituents, and the features of the dialogue units are enclosed in square brackets: ‘*f*’ refers to communicative functions, ‘*x*’ to tasks, and ‘*s*’ and ‘*p*’ are the speaker and the partner, respectively.

Move can be followed by an optional Development by the partner; this is needed e.g. in cases where the partner does not give a direct answer but initiates a clarifying sub-dialogue. After such a Development, however, the partner must return to the original topic and give the requested answer, i.e. although an optional Development is initiated, the response Move is to be found as well. The last rule describes a Development as an Exchange optionally followed by another Development. The separate structural unit Development is needed to cope with dialogue expansions which differ from the parent Exchange either with respect to the task $(x, x2, x3)$ or communicative function $(f, f1, f2, f3)$.

Discussion

Dialogue grammars can be used to describe constrained communication situations like classroom conversations (Sinclair & Coulthard, 1975) or systems giving flight information (Bilange, 1992), and it is obvious that they are useful in computational applications as providing an easy-to-implement and efficient means of restricting search space. However, they force dialogues into fixed structures consisting of simple exchange patterns so that, to be well-formed, the speakers' contributions must fit the structure. For instance, in the dialogue 6 below, the structural position of the user contribution User3 is problematic because of its multifunctionality. It can be analysed as a reaction to the list of restaurant types given by the system in System2, but at the same time, it also functions as a response to the system question in System1, giving the requested location.

- (6) User1: I'm looking for restaurants.
 System1: In which area?
 User2: What types of restaurants do you list?
 System2: Indian, Mexican, Chinese, Italian, Thai.
 User3: Try Indian in Rusholme.
 System3: Please wait....
 Here is a list of Indian restaurants in Rusholme.

If the flexibility of a dialogue grammar is increased by defining very general rules, the predictive power of the grammar is lost, and its function in the overall dialogue

management becomes questionable. For instance, the freedom to allow exchanges (and not only moves) as responses, as is done in SUNDIAL, does not necessarily make the analysis of the dialogues easier, and it is problematic what is in fact outlawed by such a general grammar. In PLUS, on the other hand, the feature values assigned to structural units needed to be specified with respect to belief reasoning and task determination, and thus the structure actually appeared to be a complex way of encoding dialogue history.

Moreover, it is not possible to reason about why moves in the exchanges are what they are or what purposes they fulfil in the structure. In order to handle miscommunication or deviations from the predefined dialogue structure, other means must be evoked. For instance in SUNDIAL, three further rule types had to be introduced to enable the smooth and correct dialogue management.

Dialogue structure grammars have been criticised also from a theoretical and methodological point of view. Their descriptive soundness is challenged because of the obvious diversity of human conversations. Moreover, the nature of conversation is not a structural product, but rather an interactive process (Levinson, 1983, p. 294), and thus the structure is created as the dialogue goes on rather than being ‘there’ for the participants to comply with. This view is also taken in Communicative Activity Analysis (Allwood, 1976), where communication is cooperative interaction with joint purpose and mutual consideration between rational agents. We elaborate this approach to ‘Constructive Dialogue Management’ in section 5.2.

3.2.3 Conversation Analysis

Conversation Analysis, developed by Sacks et al. (1974), is a research methodology for empirical study of naturally occurring conversations rather than an approach to dialogue management. It is based on ethnomethodology and emphasises the study of everyday dialogues which may have no obvious purpose at all. The conversational ‘rules’ that have been formed to describe regularities should not be understood as prescriptive rules, but rather as preferences and expectations of what will follow. The

organisation of conversations can be divided into two kinds: local organisation which deals with phenomena such as turn-taking and adjacency pairs operating across two turns only, and overall organisation which deals with sequences of turns and the organisation of conversation into a specific kind of unit.

As pointed out by Cawsey (1993), it is unclear how much of the Conversation Analysis type of research can be used in the design of human-computer interaction, although some of its insights are directly applicable and provide a useful basis for dialogue research in general. For instance, concepts like ‘turn taking’, ‘adjacency pair’, ‘(self-)repair’, ‘insertion’, ‘side sequence’, ‘opening’ and ‘closing sequence’, are generally used in NLP. A collection of papers on how Conversation Analysis can be applied to NLP is given by Luff et al. (1990).

3.2.4 Evaluation of the approaches

While the common view underlying all three approaches is that language is purposeful activity, there are differences: the intention-based approach emphasises the plans and intentions of the participants, the structural approach stresses the systematic structure of dialogues, and Conversation Analysis concentrates on empirical studies of conversational resources. The approaches are not mutually incompatible, however, but complementary: formal intention-based approaches need to pay attention to the collaborative nature of interaction and to the global structuring of dialogues, while the structural approach should take into account the participants’ goals. Both approaches should be developed on the basis of empirical studies.

The structural approach is a reasonably fast and straightforward way to model dialogues with well-defined goals and a rather straightforward global structure. Its drawback is its rigidity: once built, a dialogue grammar allows only dialogues which conform to the rules. Even if the rule system is sophisticated and rather detailed to capture various kinds of possible dialogue continuations, the approach can be criticised from a theoretical point of view in that its theoretical basis is invalid. Dialogue structure is something that appears as a result of cooperative activity by the dialogue participants,

or, as Gazdar and Mellish (1989, p. 386) claim, the syntax of discourse is “epiphenomenal and plays (at most) a subordinate role to the activity that drives the discourse.”

The intention-based approach provides a more flexible framework for dialogue analysis since it operates on the relations between successive contributions rather than on a predefined structure. It also provides a well-defined formalism to express and operate on beliefs and intentions, and it captures the logic-based model of speech acts and inferences. However, as discussed above, speech act recognition in terms of the acts’ preconditions and effects exhibits the same kind of rigidity as the structural approach: structural units (consisting either of the agents’ beliefs and goals or of plan schemata) must be used in accordance with the existing definitions. A liberation from this is to abandon the explicit recognition of speech acts and to refer to more general principles of communication instead. The approach also suffers from the same deficiency as logic-based approaches to natural language analysis in general: the neat analysis of a small NL fragment is smothered by the nasty details of a larger fragment, either undermining the basic assumptions or increasing the number of different variables.

The two approaches to dialogue management seem to overlook the rational activity that dialogue participants show when they adjust their communicative behaviour in the changing dialogue context. We accept the instrumental view of language as a means to change the context, and consider communicative acts as events that bring the contextual changes about (context referring either to participants’ mental states or the states of the world). However, we also emphasise communication as a joint activity. Dialogue are managed *constructively* to fulfil the goals that both partners have set for the dialogue.

Cawsey (1993) describes different approaches to dialogue organisation and she claims that there is no general over-arching theory of human interaction, providing a comprehensive model of how dialogue works, but that different insights can be gathered from different areas, providing a useful basis for dialogue management. We claim that there exist theories, or at least comprehensive views of communication, like CAA, that attempt to account for the whole of human communication, and that dialogue management can be based on the general principles of rational behaviour.

3.3 Natural language generation

This section reviews some basic notions in the planning of system responses and generation. Especially, we study cooperativeness with respect to the system's over-answering capability, and discuss three different types of system responses which have been subsumed under the notion of cooperativeness in the literature: those based on correcting the user's presuppositions (Kaplan's (1983) COOP and Wahlster et al.'s HAM-ANS), recognising the user's plan (Allen, 1983) and preventing the user from drawing false implicatures (Joshi et al., 1984). We aim to bring these different aspects together with the help of pragmatics-based dialogue management in Section 6.4.

3.3.1 Introduction

The task for generation in general is to define how surface level utterances arise from communicative goals in a specific discourse context. In the literature, the planning of system responses is usually divided into two levels: strategic, or 'what to say' and tactical or 'how to say it'. The distinction was originally made by Thompson (1979), but has also been referred to as content planning and realisation, deep vs. surface generation, or text planning and plan execution. This seems intuitively plausible and useful when referring to special problems encountered in the generation task. It is widely used in generation systems, where the distinction corresponds to two components which use different data structures and inference mechanisms. However, it is also widely acknowledged that this division is problematic, since the form and content of surface level utterances depend on both the communicative goal and linguistic resources, see e.g. Appelt (1985), McDonald (1983), Hovy (1986, 1988c), Danlos (1987), Kantrowitz and Bates (1992), Rubinoff (1992). Meteer (1991) has referred to this as the 'generation gap'. Moreover, even if the distinction may be useful on a conceptual level, its implementation is difficult because of the various ways in which planning and realisation can interact. An interleaved architecture where the two components can interact has been advocated, to address the problems of planning and generation in a more flexible way: the system can adjust its state according to changes in the goal,

and this approach also permits the incremental production of output (Hovy, 1988c; Kantrowitz & Bates, 1992; Rubinoff, 1992).

Some generation systems like PENMAN (Mann & Matthiessen, 1983), exploit networks to represent the necessary information for planning, combining the separate levels of planning into an integrated traversal of interrelated network systems. However, network traversal requires some specification of how alternative paths will be selected. PENMAN specifies preselections for the alternatives by means of Sentence Plan Language expression which is produced by an application-specific module prior to the network traversal. The strategic vs. tactical ‘generation gap’ is not avoided.

Recently, a third question for generation, ‘why’ has drawn more attention. The question was raised by Hovy (1988a), and the system’s ability to reason about why it chose to say what it did has subsequently been addressed as an important issue in natural language generation (e.g. Moore & Paris, 1993).

Another distinction in planning is domain or content planning vs. discourse planning (e.g. Litman, 1985; Litman & Allen, 1984, 1987): knowledge of the speaker’s goals (domain plan or knowledge of the task) and communicative knowledge (discourse plan) are separated. Two-level planning has been exploited among others by Murray (1989) in planning curricula and by Cawsey (1990, 1993) in explanation generation. In PLUS this distinction was reflected in the division of the system goals into task-oriented and dialogue-oriented goals; the name of the first goal type is unfortunate however, as the planning deals not only with tasks to be completed, but also with the information that needs to be communicated to the user. Grosz and Sidner (1986) point out that speaker intentions must be distinguished from thematic coherence or focussing structure, since the same intentional structure can give rise to different attentional structures in different discourses.

As in dialogue management, two approaches can be distinguished in natural language planning and generation: structural or schema based planning (e.g. McKeown, 1985), and discourse relation based planning (e.g. Hovy, 1988c, 1990). The former assumes that texts have a stereotypical structure which can be captured and precompiled into

schemas. The schemas then govern the content of a text paragraph as well as the order in which the material is presented, and thus implicitly encode the coherence requirements of the text. The absence of internal structure in the schemas has lead some researchers to consider whether a coherent text structure can be obtained by defining a set of basic relations that hold between text sentences. Planning based on discourse relations has mostly used Rhetorical Structure Theory (RST), developed by Mann and Thompson (1987) for descriptive purposes. The problem with RST is that the rhetorical relation definitions are underspecified and the RST analyses are inherently ambiguous: different rhetorical relations may hold between two sentences. It may not be obvious which one is the intended one. Moreover, it may be that the relation indeed conveys several rhetorical intentions at the same time: there is no single well-suited RST relation to describe the sentence connection. There is no conception of the speaker's intentions in pure RST, and thus it cannot address questions concerning why a particular relation has been chosen unless the relations are augmented with information about the speaker's goals (cf. Moore & Pollack, 1992; Moore & Paris, 1993).

There are also attempts to generate coherent texts without trees. Sibun (1991) describes the Salix system which relies on the domain structure of the subject matter to generate texts incrementally. The text is locally organised and its structure is the structure of the domain: the coherence of the text is thus solely derived from the domain coherence. Sibun's approach is close to our 'Constructive Dialogue Management' in which the system responses are planned in the local context in the process of evaluating the user contributions with respect to the principles of ideal cooperation, see Section 5.2.

3.3.2 Cooperativeness in Response Planning

Cooperativeness and presuppositions

One of the earliest works on cooperative response planning, Kaplan's COOP system (Kaplan, 1983), aimed at giving a relevant response in case the answer to the user's question was "no". If the presuppositions of the query were not true, the system would come up with an answer where these were corrected. For instance, if the course CS121 had not been given, the simple answer 'None' in the following dialogue would be uncooperative, since it would lead A to believe that the course CS121 was very easy or that the students were very bright. In other words, it allows A to draw false implicatures on the basis of false assumptions.

- (7) A: How many students failed the CS121 last term?
B: a) None.
b) The course CS121 was not given last term.

Another early system where cooperative responses were studied was HAM-ANS by Wahlster et al. (1983). As pointed out by Webber (1987), their approach complements that of Kaplan's in that they aim to give additional information beyond that requested in cases where the answer is positive. For instance, in the following dialogue, a follow-up question about the location is anticipated by giving that information in the answer.

- (8) A: Has a yellow car gone by?
B: Yes, one yellow one on Hartungstreet.

Cooperativeness and plans

A similar type of 'over-answering' was also a feature in the work by Allen (1983), where the additional information is linked to the recognition of the user's plans and the system's attempts to remove obstacles from the execution of these plans. For instance, if the user is to board a train and asks for the departure time, the system will also give information about the gate where the train leaves, since not knowing the departure location may prevent the user from executing her plan.

- (9) A: When does the Windsor train leave?
B: 4 pm., Gate 7.

Cooperativeness and false implicatures

Finally, Joshi et al. (1984) link cooperativeness to Joshi's (1982) revised version of the Gricean maxim of Quality:

If you, the speaker, plan to say anything which may imply for the hearer something that you believe to be false, then provide further information to block it.

Joshi et al. (1984) contrast their work with previous work in that the user's expectations about the information provider's cooperative behaviour are taken into account: e.g. the expert is assumed to come up with a better plan, if the recognised one is suboptimal or unsuitable. The additional information is already available as a side-effect of the search which the system needs to do to give an answer to the direct question. The system's task is to block unwanted implicatures by checking if the direct answer enables the user to draw conclusions which the system knows are not true. This can be done by using default reasoning, and restricting relevant knowledge to the subset of the knowledge base which is in focus because it provides the direct answer to the user's query.

3.4 Conclusion

In this chapter we reviewed three issues that we consider pertinent for the purposes of the thesis: pragmatics, dialogue management and natural language generation and planning. We advocate the view that pragmatics is necessary in the design and development of natural language dialogue systems, and specify cooperativeness, coherence and robustness as key issues in pragmatics-based dialogue management. Furthermore, we understand pragmatics as the study of the agents' communicative competence, and thus, improving the system's communicative competence means improving the ade-

quacy of the system responses as well as friendlier and more natural dialogue behaviour in general.

We discussed two current approaches to dialogue management: Intention-based and Structural approaches, and concluded that the approaches seem to overlook the general context in which the dialogue takes place as well as the dialogue participants' rational activity. We will propose the Constructive Dialogue Management approach as an alternative in the Part II of the thesis.

We also discussed natural language generation and especially different aspects of cooperative response planning: cooperativeness related to presuppositions, plans and false implicatures.

Chapter 4

Communicative Activity Analysis

In this chapter, we review Allwood’s ideas of communication which have served as a source of inspiration for the thesis. We will call this approach ‘Communicative Activity Analysis’ (CAA), after Allwood and Haglund (1991) who analysed PLUS dialogues in these terms. CAA is a pragmatic-philosophical approach to communication in general, rather than a formalised and fully developed theory, and its broad view-point aims to cover all aspects and levels of different types of communication. Expounded by Allwood in his inaugural dissertation (Allwood, 1976), this approach has been gradually evolving in the course of time, although the basic insights of communication and rational agenthood have remained the same.

The CAA approach has been used in studies of adult language acquisition (Allwood, 1993), Wizard-of-Oz dialogues (Allwood & Haglund, 1991), and linguistic feedback (Allwood et al., 1992). A situation theoretic analysis of Allwood’s communicative sender activity has been presented by Nivre (1992b). The current thesis aims to present a CAA-based approach to computational dialogue management and system response planning. The way in which it has been implemented in our work is motivated by our goal of studying human-computer interaction and robust dialogue management. In other words, the formalisation of CAA goes as far as it serves the purpose of building a cooperative and robust dialogue manager capable of producing coherent dialogue contributions. This is in accordance with Allwood and Haglund (1991) who emphasise

that the designers of service systems should use CAA as appropriate to the system: it is not necessary to formalise the whole of CAA, since the system does not necessarily need to reason about all aspects of robust communication, in order to behave in a certain robust way.

4.1 Levels of Communication

In a series of papers (Allwood, 1976, 1978, 1992; Allwood & Haglund, 1991), Allwood has been developing ideas about human communication as cooperative activity between rational agents. According to him, human communication creates normative social obligations which deal with the participants' evaluation of whether and how they can and want to continue the dialogue, perceive and understand the partner's contribution, and react to them in some attitudinally appropriate way. The agents act in accordance with their own goals and they are also capable of attending to the other's goals and abilities. Communication is thus seen as an instrument which serves individual and collective information processing, and it is a part of social activity whose main purpose need not be communicative.

There are several levels of organisation in human dialogues, each of which provides necessary, but not sufficient enablements and constraints on the communication. CAA lists the following levels (Allwood, 1992):

1. **Physical:** communicators are physical entities and communicative contributions are physical processes,
2. **Biological:** communicators are biological organisms and communicative contributions are directed behaviour,
3. **Psychological:**
 - (a) **Perception, understanding and emotion:** communicators are perceiving, understanding and emotional beings and their communicative contributions are perceptually comprehensible and emotionally charged phenomena.
 - (b) **Motivation, rationality and agency:** communicators are motivated, rational agents, and their communicative contributions are motivated rational acts.

4. Social:

- (a) **Culture, social institution, language:** communicators are members of a culture, social institution and linguistic community and their communicative contributions are cultural, social institutional and linguistic acts.
- (b) **Activity:** communicators play certain roles in social activity and their communicative contributions are contributions to that activity through their role.
- (c) **Communication:** communicators are normally primarily either in the sender role or in the receiver role, and their communicative contributions can thus be characterised by such labels as: sent message, speech, writing, statement, question, and request.

In human-computer interaction, the initial setting is of course more limited than in human-human communication, and the levels mentioned above may appear too wide. However, physical or biological levels are not irrelevant, although they have been overlooked in NLP work: the speed of the system, the clarity and movement of the cursor, the clarity of the text and window system, as well as the user's vision, capability of typing, handling the mouse etc., are important factors in making the human-computer interaction convenient.

As for the psychological and social levels, they have generally been acknowledged as forming an important part of the context which NLP must address. For instance, according to Hovy (1986, 1988a), the pragmatic information needed in NLP applications comprises interlocutors' personal characteristics, the speakers' goal with respect to the hearer, and the conversational atmosphere (cf. p. 30). The importance of this wider context has also been pointed out by Tsujii et al. (1992, p. 63), who emphasise the role of global context in their sublanguage-based approach to NLP: the context in which texts are prepared and conversations take place is established by 'communicative environments' which include aspects like types of writers, readers, level of formality, and topic domain.

In this general setting for the organisation of communication, Allwood studies linguistic communication as rational and cooperative activity. He introduces seven principles that capture the meaning of 'normal, rational agenthood' (Section 4.2)¹, and defines

¹Section numbering in this paragraph refers to this thesis, not to Allwood (1976).

‘ideal cooperation’ with respect to these principles, emphasizing especially its ethical aspects (Section 4.3). He then considers basic requirements for communication and especially the communicative activities of the sender and the receiver, and defines the type of communication that is typical among human beings, ‘full-blown cooperative communication’, as ideal cooperation between normal rational agents (Section 4.4). Communication creates normative social obligations, in terms of which communicative responsiveness can be defined (Section 4.5). By participating in the communicative activity, participants have certain roles which further determine their communicative activity (Section 4.6). Finally, communicative acts are derived from general communicative principles, and can be said to carry expressive and evocative dimensions (Section 4.7).

We discuss CAA from the point of view of what we consider relevant in our own research in human computer interaction, and want to emphasise that some aspects of Allwood’s work have not been considered at all (e.g. conventions and conventionalisation, nor the thorough albeit philosophical discussion on sender and receiver activities).

4.2 Rational Agenthood

According to Galliers (1989), Allwood’s work is unusual in that it is one of the few attempts to summarise the traits of a rational agent. Allwood (1976) summarises these traits in seven principles that concern normality, rationality and agenthood. All the principles can be understood as “statements of norms that an individual agent tries to follow in his own behavior”, or “statements of assumptions that typical socialized agents make about the behaviour of other individuals.”

1. **Normal rational agenthood:** “Typical human beings are normal rational agents”.
This is the precondition for the other principles.
2. **Agenthood**
 - (a) **Intentional and purposeful behavior:** “The intentionally controllable behavior of an agent is intentional and purposeful”.

This principle postulates that behaviour should have a “point”, i.e. the agents should have a reason for their actions.

- (b) **Voluntary action:** “The actions of an agent are not performed against his own will”.

Since there is a trivial sense in which every action is voluntary, the negative formulation is used to emphasise the stronger sense of “voluntary”, according to which an action is voluntary if and only if an agent does not believe that she would be in danger if she does not perform the action. In this stronger sense of “voluntary”, Allwood can distinguish involuntary action from voluntary, an example of the first one being forced labour at gun-point in a concentration camp.

3. Normality

- (a) **Motivated action:** “The actions of a normal agent are motivated”.

Motivation here concerns a need, desire, wish or want, but excludes the external circumstances that give rise to the motive.

- (b) **Pleasure and pain:** “Normal agents do not act so as to decrease their pleasure and increase their pain”.

The negative form relaxes the principle, as the positive formulation “agents strive to increase their pleasure and decrease their pain” is considered too strong.

4. Rationality

- (a) **Adequacy:** “The actions of a rational agent are selected so as to provide the most adequate and efficient way of achieving the purpose for which they are intended”.

- (b) **Competence:** “The actions of a rational agent are performed only if he thinks it is possible to achieve their intended purpose”.

Intentional and purposeful behavior, motivated action and rationality are principles which can be directly applied to human-computer interaction. We spell them out as rules which govern the system’s reasoning: determining joint purpose and checking the system’s communicative obligations.

4.3 Ideal Cooperation

Participants are engaged in ideal cooperation, if the following constraints are fulfilled:

1. joint purpose,
2. cognitive consideration,
3. ethical consideration,
4. trust.

Joint purpose refers to the fact that the participants have common purposes that they voluntarily strive to achieve. Allwood seems to assume that the participants share common goals, at least on the very general level of transfer of information. Although cooperation is considered the fundamental characteristic of linguistic communication, this does not mean that the communicators fully cooperate all the time: sometimes cooperation is only partial and the agents pursue their own private purposes. However, as long as the agents are communicating, they have at least one common purpose: transfer of information. According to Allwood, this purpose is the basis of communication and if it disappears there will be no communication. This idea forms the basis for the response planning which we will describe in Chapter 8: the planning starts from ‘NewInfo’ which encodes the new information to be communicated to the user in the dialogue context. If there is no new information to be exchanged, there is no need to communicate either.² To determine joint purpose for a dialogue is an important part of the system’s reasoning, and it will be discussed in detail in Section 5.2.3.

The two consideration constraints form the basis for evaluating the contributions. Cognitive consideration deals with the participants’ perceptual and cognitive activity: participants evaluate the contributions on the psychological level and try to predict actions

²Here we have excluded the *phatic* function of communication which deals with the establishment and maintenance of communicative contact, see Lyons (1977, p. 53); the term is originally coined by Malinowski (1923). However, if we understand ‘information transfer’ widely, we can say that the new information in phatic communication equals the management of social relations, opening up and keeping open the channel.

and reactions of the partner, etc. Ethical consideration is the main contribution of Allwood's theory to the discussion of rational cooperation. While cognitive consideration guarantees the agent's awareness of another agent's goal, ethical consideration requires that the agent should not do anything that would prevent the other agent from acting as a rational, motivated agent. In order to show cooperation it is not enough that the partner's goals are known: the partner is also to be treated ethically.

The last constraint on ideal cooperation is trust. In order for the cooperation to work, the participants must trust each other to act according to the other three constraints.

4.4 Full-blown Communication

Communication is related to the sending and receiving of information from one agent to another. In order for the agents to communicate in the first place, some basic requirements must be fulfilled. Allwood distinguishes four necessary requirements for communication:³

1. contact,
2. perception,
3. understanding,
4. attitudinal reactions.

The first two requirements are actually preconditions for the latter two which involve higher-level intentional reasoning. They need not be fulfilled, however, as in situations where communication takes place without contact (e.g. with TV presenters) or perception (unintended communication). Allwood (1976) discusses different types of communication, and defines typical human communication as a basically cooperative activity, where all the requirements are fulfilled. This type of communication is called

³Allwood (1992) calls these four requirements 'communicative functions' which is unfortunate because of the ambiguity of the term with respect to speech act theory.

‘full-blown cooperative communication’. When talking about ‘communication’ in this thesis, we refer to ‘full-blown communication’.

If the dialogue participants act according to the principles of ideal cooperation, they must, after each contribution, evaluate whether the requirements for communication are still valid: they must consider how they can and want to continue, perceive and understand the contribution, and react to it. However, they can respond in different ways depending on which level of requirements they want to respond to, or whether they want to respond positively or negatively, explicitly or implicitly. They can give *feedback* on the evaluation on different levels (Allwood et al., 1992).

Communication is an interactive task. Therefore, each communicative contribution also serves communication management purposes which aim to maximise the following general goals: freedom of communication, avoidance of pain and possibility to seek pleasure, and correctness of information (Allwood, 1976). Allwood and Haglund (1991) make these goals less abstract by dividing communication management into two kinds: the speaker’s own communication management and interactive management. The speaker’s own communication management deals with the speaker’s on-line planning: correcting and changing one’s contribution while communicating. Interactive management concerns turn taking (distribution of the right to speak), sequencing (structuring the dialogue into sequences, subactivities, topics etc.) and feedback (elicitation and giving of information about the four basic communicative requirements). We will not deal with the speaker’s own communication management in this thesis. Interactive management is discussed in Section 5.2.5.

4.5 Communicative responsiveness

As mentioned above, communication creates normative social obligations, which deal with issues such as people’s availability and contactability for information coordination and their willingness to report on the results of the evaluation of the information. These obligations can be understood as *obligations of responsiveness*, where responsiveness is

itself a consequence of the human ability for rational coordinated interaction (Allwood, 1992).

Communicative responsiveness can be associated with what have been called ‘preferred responses’ (Schegloff & Sacks, 1973) or ‘preferreds’ (Levinson, 1983) in Conversation Analysis. This means that a communicative act often puts a certain pressure on the hearer to react in a certain way. For example, an initial greeting gives rise to a pressure to respond with another greeting, and an expression of gratitude to respond with a disclaimer. However, as claimed in Allwood and Bunt (1992), these concepts do not capture the normative aspect of responses which underlies communication in the first place. It is the obligation of responsiveness which requires the hearer to evaluate the utterance and respond to it in an appropriate, socially committing way.

Responsiveness is an essential part of being cooperative: being able and willing to respond is shown in the dialogue participation by appropriate responses. It is combined with other obligations which affect the form and content of the response. In particular, it is tied to the activities and roles that the dialogue partners are engaged in. For instance, one is under different pressure when answering the questions of one’s manager than when answering the questions of a nosy friend. The employee role thus reinforces the responsiveness obligation in relation to the manager.

We can now also describe what constitutes the ‘answerhood’ of an utterance in general. A question usually requires a response, but the response need not be a direct answer. Levinson (1983, p. 293) points out that a question like *What does John do for a living?* can be happily followed by

- partial answer (“Oh, this and that”),
- rejection of the presuppositions of the question (“He doesn’t”),
- statement of ignorance (“I’ve no idea”), or
- denial of the relevance of the question (“What’s that got to do with it?”).

What makes an utterance after a question function as an answer is that it occurs after the question with a particular content. In CAA terms, we say that the response

is evoked by the obligation of responsiveness, and its functioning as an answer to a question is based on the four requirements of ideal cooperation, especially on the hearer's trust that the partner replies in a rational considerate way. The relevance of the response in its context is evaluated by the hearer according to the communicative obligations and knowledge about the context and world.

It is important to make the following terminological distinction, applied in the thesis: an 'answer' is a contribution which attempts to satisfy the evocative intention of a 'question'. A 'response' is a wider notion and refers to "any contribution following a question".

4.6 Roles and activities of participants

By virtue of participating in activities, the participants occupy or play roles which are constituted by the activities. The roles can be characterised by global communicative rights and obligations that the participants have in different dialogue situations, and the participants can have several roles and their roles can also change. The information giver has different obligations from the information receiver, and the roles are further differentiated when e.g. asking a stranger how to get to the airport or asking one's friend what would be the best way to go to the airport.

From the user's point of view, interaction with an information-providing system is a social activity in which the user is engaged: in order to perform a certain task, she needs to obtain information which requires communication with the system. For instance, she may be planning a journey which involves the use of a hire car, and thus she seeks for information on hire car companies. The user is thus typically involved in two related activities: communicative activity with the information-providing system and non-communicative activity with respect to the task which the dialogue serves. As mentioned in Section 3.3.1, these two related activities have been called discourse level and content level (domain) planning (Litman, 1985; Cawsey, 1993).

In principle the user has no communicative obligations when engaged in an information-

seeking dialogue with the system: she can choose not to initiate a dialogue or to terminate a dialogue at any point. However, if she plays the role of an earnest information-seeker, the dialogue constitutes a situation where she is constrained by local communicative obligations that are related to her being a cooperative rational agent. These obligations include willingness to provide the information (specification, clarification) requested by the system to fulfil the search task, and willingness to terminate the dialogue explicitly.⁴ On the other hand, the user has the right to continue requesting information as long as she likes – and is willing to pay.⁵

The system's role in an information seeking dialogue is to provide truthful information, but not necessary advise or assist the user in her domain planning (see van Loo and Bego (1994) about different agent tasks in information-seeking dialogues, and how dialogues can be classified according to the agents' expertise and involvement in the planning). On a general level, this means that the system is always ready to accept user contributions as input, produce their analysis, and provide the user with appropriate responses, but that the system's communicative activity is restricted with respect to introducing new topics. The system described in this thesis has a simple task structure: request and provide information, and thus we will not go into details of task planning. However, this simple task structure allows us to concentrate on the system's communicative obligations and model the collaborative nature of dialogues by the system's ability to evaluate the user requests with respect to its own goals as an information provider. Thus the system not only gives simple answers to the user's requests, but is also allowed to take an initiative to clarify or specify the user request, and especially, pursue the goals that it has adopted until the user explicitly denies these (see Section 5.2.6).

Figure 4.1 summarises the role analysis of the system's rights and obligations as formulated for PLUS (taken from Allwood and Bunt (1992)). These general guidelines

⁴The obligation of explicit closure of a dialogue is not supported by the corpus studies, see Section 2.3.2. However, the lack of support is most probably due to the unnatural setting of the dialogue situation, and not an indication of the irrelevance of the obligation.

⁵The aim in PLUS was to build a dialogue system which would enable the user to access the Electronic Yellow Pages via her home computer screen.

The system's global obligations: When open, the system must, unless it is overoccupied:

1. always start by greeting and offering factual information service.
2. always be willing to continue the dialogue.
3. always try to receive and understand the user's contributions.
4. if the system cannot continue, it must explain why.
5. if the system cannot perceive or understand the user's contribution, it must attempt to get the user's help for clarification.
6. always try to provide the information the user is requesting (this implies relevant and true information).
7. always present available information in an optimal way for the user, i.e., perspicuously and in such a way that the information given is not too much (and not too fast or too little or too slow).
8. if no information is available, offer the user other information or advice on where to get other information.
9. if too much (too little or unperspicuous) information is available, take measures to make presentation more optimal for the user.
10. never terminate a session unless the user has explicitly given a termination signal or not responded within 1 minute to the system's request concerning whether the user has further queries.
11. if the user terminates explicitly give an offer of renewed service.

The system's rights: The system has no clear communicative rights.

Figure 4.1: The system's communicative obligations and rights. Taken as such from Allwood and Bunt (1992).

are also assumed in the system described in this thesis, except that the right to try to pursue its own goals is added. This allows an exploratory study on the evaluation of conflicting goals by the system.

4.7 Expressive and evocative dimensions of communication

Each communicative act carries both expressive and evocative dimensions. The *expressive dimension* deals with the expression of an attitude on the part of the speaker, and the *evocative dimension* with the evocation of a reaction in the listener. By the expressive content of a communicative act the speaker gives information about her mental state (emotions and attitudes), but also about her physical and social identity. By the evocative content of a communicative act the speaker intends to influence the hearer's mental state, and at very least wants the hearer to apprehend some information. Allwood also distinguishes the third dimension, obligative dimension, to capture the social commitments connected with communication. This is related to performative utterances. In the later works, Allwood seems to conflate obligative dimension with the speaker's obligations in general, and this is the way we consider it in the thesis.

As argued in Allwood (1978), these functions must not be confused with the illocutionary and perlocutionary forces of speech acts. Rather, Austin's concept of illocutionary force is split up into three dimensions of expression, evocation and obligation, none of which need be conventional, and his notion of perlocution corresponds to what is actually achieved by a communicative act (evoked response). Illocution is seen as what is given by the content of the so called "communicative activity verbs" ('say', 'baptise', 'forbid'), and the conventional content of illocutionary acts is reduced to the conventional content of locutionary acts (which includes the sentence mood). The effects of an utterance and the intentions behind them are also distinguished. This distinction corresponds to that between the actually evoked response and the evocative intention of an utterance, respectively.

Table 4.2, taken from Allwood (1992), summarises stereotypical expressive and evocative functions of the four conventional communicative acts.

The main evocative dimension of the contribution is directly mapped from the surface representation: the speaker either wants to elicit information or give information. This

Type of communicative act	Expressive dimension	Evocative dimension
Statement	belief	(that listener shares) the belief judgement
Question	desire for information	(that listener provides) the desired information
Request	desire for X	(that listener provides) X
Exclamation	any attitude	(that listener attends to) the attitude

Figure 4.2: Expressive and evocative dimensions of four communicative acts. From Allwood (1992).

should not be confused with the recognition of speech acts, however, since the exact nature of the communicative act is based on the rationality considerations of what each partner is attempting to achieve via their engagement in the dialogue.

4.8 Conclusion

In the following chapters we use CAA to study how aspects of robustness, induced from the corpus, can be incorporated in a dialogue manager and how the system's general communicative principles can be modelled. Of course, we must emphasise that it is not necessarily desirable or even possible to formalise the whole of CAA in order to build a simple service system. Some dimensions of rational agenthood and full-blown communication do not become realised in human-computer interactions (e.g. voluntary action and motivation to seek for pleasure and avoid pain), while some factors that affect the system's communicative competence are constant and the system does not explicitly reason about them (e.g. in our prototype the roles of the user and the system as an information seeker and information provider). Communicative competence can also appear as one of the system's procedural features and be 'hidden' in the inferences that the system draws in order to provide appropriate responses. This is especially true of 'cooperativeness': it is not modelled by a single rule which explicitly

encodes the system's desired behaviour, but rather, it emerges from the global system design during the reasoning processes. The balance between what should be stored as rules that govern the system's inference processes and what should be exhibited by a robust and helpful system as a whole, is a decision that depends on the level of generality that the system aims at. However, it is important to have a general theory of communication, in order to decide which factors of cooperative communication are relevant and need to be taken into account on any chosen level of generality.

Part II

Pragmatics-based Dialogue Management

Chapter 5

Constructive Dialogue Management

From the point of view of system design, dialogue management should be based on both a theoretical foundation in a theory of communication and an empirical foundation in a corpus of typical dialogues that the system is to handle. System requirements should mirror some standard of appropriate behaviour in a given situation, and adequacy of the system responses should be defined with respect to dialogues that naturally occur in that situation. The characteristics should be abstracted on a level where their formalisation within some theoretical framework can be done. Finally, this model should be implemented by choosing appropriate data structures and control mechanisms.

This chapter presents the two first tasks for designing dialogue systems: desiderata drawn from empirical research and the formalisation of the desired features. We discuss ‘robustness’ of the system on the basis of Wizard-of-Oz dialogues collected in the PLUS project (see Section 2.3), and associate it with the system’s communicative competence. To formalise the robustness features identified from the corpus we refer to pragmatic principles, using Allwood’s (1976) theory of communication as cooperative rational action. Our investigation is restricted to information-seeking dialogues, i.e. dialogues which are initiated in order to get information about some particular topic.

5.1 System robustness and dialogue studies

Based on our corpus studies, we have identified four general requirements for dialogue systems:

1. physical feasibility of the interface,
2. efficiency of reasoning components,
3. ability to cope with syntactically problematic input,
4. ability to initiate responses that clarify vagueness and misunderstandings.

All four requirements can be based on the CAA principles, and thus they are not just separate items in a list but different aspects of ideal cooperation and rational motivated activity. The first two are not directly addressed in the system presented in the thesis, but the last two are considered important design principles.

5.1.1 Physical requirements for the user interface

As argued in Section 4.1, human-computer communication has constraints on physical and biological levels of communication and is thus influenced by factors like the user's eye-sight and physical capability to type or use the mouse. Although the effect of these factors is not studied in this thesis, we want to point out that it is directly related to the basic requirements of full-blown communication: contact and perception. In our corpus collection studies it became evident that the user's communicative behaviour was affected by these kinds of factors: the users who were inexperienced with computers and minimally acquainted with typing and other computer interface requirements, were slow, confused and unsure about how to communicate with the computer.

In HCI, the design of flexible, simple and easy-to-use interfaces has been one of the main objects of research, and concentrated on 'human-factors' as well as providing software tools, see Wallace and Anderson (1993). Burton and Steward (1993) noticed that extralinguistic editing facilities can actually replace linguistic enhancements like

the use of ellipsis in queries (see below). However, we should bear in mind that for some potential users typical mouse-handling operations like cut and paste may turn out to be extremely difficult physical tasks. A natural language interface can thus provide a user-friendly way to communicate with the computer, given that the system is robust and capable of handling fragmentary, sometimes misspelt input. Of course, combined with other modalities like graphics and speech, a natural language interface is a powerful tool in human-computer communication. Multimodality is an area of growing interest, but more research based on a general theory of communication is needed to show how robust interfaces can be built.

5.1.2 Efficiency

In the PLUS Wizard-of-Oz studies, even when subjects acknowledged the naturalness of the system responses, they regarded the long response time as the main factor in characterizing differences between their computer conversations and real human-human conversations. Moreover, efficiency of reasoning contributes to the contact requirement: if the system does not reply in a reasonably short time, the user normally starts to think that contact with the system has been lost.¹

Considerable attention should thus be paid to fast and efficient algorithms in building robust systems. In the system presented in this thesis, however, this aspect is not demonstrated: it is not our aim to develop computationally efficient algorithms for software development, but a robust dialogue model which exploits pragmatic communication principles. The study of efficient implementation possibilities is one of the future directions of the work.

¹In the French dialogues collected in PLUS, the long response time also caused unnecessary turn takings and complications in the interactive dialogue management.

5.1.3 Syntactically problematic input and output

Instead of resembling an intelligence test on computer skills or an exploration of the system's limitations, an NL interface system should be easy to 'talk to'. This idea is advocated in HCI research (e.g. Wallace & Anderson, 1993), and is referred to as the 'habitability' requirement in Ahrenberg et al. (1994). However, linguistic sophistication in an NL interface has often been questioned as esoteric and rather unimportant (Diaper, 1986; Burton & Steward, 1993; Ahrenberg et al., 1994), mainly because the emphasis has been on building a simple but working service system. For instance, Burton and Steward (1993) conducted experiments where they studied the effects of intersentential linking devices like ellipsis on the usability of an NL interface, and concluded that natural language enhancement does not necessarily improve the system's usability, but may in some cases even obscure the task.

Our contention is that a good natural language interface can indeed improve the usability of a system. This will reduce unintended problem solving situations like "what would be the best way to put the questions so that it would understand me".² However, such an interface should be based on solid pragmatic principles rather than enhancement of purely linguistic capabilities: in fact some syntactic phenomena like ellipsis, preposition attachment and coordination ambiguities require reference to the context in which the sentence occurs, as well as requests which are syntactically and/or conceptually vague. The negative results of Burton and Steward for instance were mainly due to the simple pattern matching algorithm which was used in the recognition of elliptical user questions: the understanding of ellipsis does not depend on the local syntactic context only, but on the whole dialogue context.

More extensive use of pragmatics in a dialogue system also means that the role of the natural language front-end, i.e. the parser and generator, is redefined. The dialogue manager takes care of the 'understanding' process, not the parser, and in a similar way the dialogue manager is also responsible for much of the tactical generation process

²One of our subjects reported that he had been wondering whether the computer would understand 'restaurants indian' or 'indian restaurants'.

producing a highly specified representation for the surface generator to realise. This change in the division of labour between linguistic resources and conceptual reasoning thus affects the functionality of the parser: because the dialogue manager takes care of the interpretation of the input with regard to context, the parser should deliver a parse for various kinds of corrupt input, and not necessarily one which is semantically fully specified. The generator, for its part, should be able to produce contributions which do not necessarily form a complete sentence, and may do so in an incremental way. In PLUS, this kind of new way to build natural language front-ends was supported: instead of grammars which cope with sophisticated linguistic phenomena, a ‘shallow’ grammar was advocated. The ‘shallow’ surface generator used in PLUS is described in (Black & Cunningham, 1992). An enhancement of this, designed especially for incorporation in a pragmatics-based dialogue system (Lager & Black, 1994), is also used in our system.³

5.1.4 Speaker initiatives

The fourth and last requirement for robustness deals with the conversational adequacy of the system responses. This manifests itself in the contributions that the system plans to clear up vagueness, misunderstandings or lack of understanding in user contributions. Let us consider three cases from the dialogue corpus to see how these aspects can be dealt with: clearing up vague requests with respect to the system’s world knowledge, clearing up vague requests with respect to the system’s application knowledge⁴, and responses to requests where the system limits are encountered (these aspects can be compared with those discussed in the thesis introduction (Chapter 1) for the sample

³As a step towards more flexible interfaces, we can also mention incremental processing, whereby analysis and generation are performed in accordance with the incoming data. An example of such a generator is McDonald’s MUMBLE (McDonald, 1983). From the psycholinguistic view-point, incremental generation is advocated by de Smedt and Kempen (1987). A novel type of incremental flexible generator, capable of dealing with fragmentary input in a highly interactive environment is introduced in van de Veen (1994).

⁴This is knowledge about the issues related to the activity in which the user and the system are engaged.

(restaurant type). The user supplies information about the type of the restaurant, and will eventually get the required list.

Another example of this kind of vagueness resolved on the basis of world knowledge appears in the PLUS sample dialogue. The whole dialogue is given as (1) in the very beginning of the thesis introduction.⁵

- (11) User1: I need a car.
 System1: Do you want to buy or rent one?

The user contribution is understood by the system as the user wanting to have a car, and the system, on the basis of its role as a cooperative information provider, interpretes this as the user wanting to have information on how to get a car. The world model contains the information that the user can have a car by buying, renting or stealing a car. Filtering the alternatives with respect to the application model, the system ends up with the result that it can provide information on either buying or renting a car, since the Yellow Pages has not got any information about how to steal a car.⁶ The system now formulates its own goal as a clarification of the part of the database to search. At the same time the vague user statement of needing a car is clarified, by the question *Do you want to buy or rent one?*

Both examples show how the wizard has been able to clarify a vague user request with respect to world knowledge and application model. They also show how the wizard plans the responses so that they pursue her own goal (provide information from the Yellow Pages) and at the same time address the coherence requirement concerning what is the topic of the dialogue (the user talks about the needing of a car, not about different companies that supply cars).

⁵Because the dialogue is modified from the corpus dialogues, we use 'System' instead of 'Wizard'.

⁶The system may also know that stealing is illegal and thus it must not provide any information on it.

Vague requests and application knowledge

In dialogue (12) the user supplies the keyword ‘restaurant’, but specifies it with a evaluative adjective ‘good’. The wizard wants to restrict the database search as above, but the user’s specification is of no help. The specification is understood, but the limits of the wizard’s knowledge are encountered; there is no information on the quality of the restaurants in the YPs.

- (12)
- User1: hello can you tell me where there are some good restaurants
Wizard1: The YP has no details of the quality of restaurants.
Can you be more specific as to area and/or type?
User2: i’m looking for somw spicy food in the centre of manchester
Wizard2: I have Indian, Chinese, Thai, and Mexican restaurants listed.
User3: what about an indian then
Wizard3: Please wait...
Gay Lords Restaurant 832-4866
Koh-i-noor 236-5882
Star of India 236-5798
Piccadilly Indian Restaurant 236 6256
Rajdoot 834 7029/2176
more? ...

The evaluation of the user contribution has now led to a different dialogue situation: the wizard wants to pursue her goal to clarify the user request and at the same time inform the user why her request could not be carried out directly. The response communicates both intentions, formulated as a contribution which contains two sentences connected via some sort of justification relation. The wizard has also given the user both possible specification parameters, instead of just asking one of them.

The dialogue continues by the user providing the location of the restaurant, and also the food type. The wizard does not understand the misspelt type specification, but because the type has already been asked, she prefers to give the alternatives for the user to choose from, instead of asking for the type explicitly. The response is conversationally more adequate than a simple repetition of the type question, since it also conveys some new information to the user about how to specify the type so that the partner understands it.

The adjacency pair (Wizard2)-(User3) is an interesting example of non-conventional

encoding of communicative acts. The acts ‘request’ and ‘inform’ are expressed in contributions whose literal force is exactly the opposite: the wizard asks the user to select the restaurant type by providing a list of alternatives in a declarative statement, whereas the user selects one by an (elliptical) question.

In this dialogue, the wizard’s communicative competence is shown in her providing information about the application (and not just from the application). Thus she can help the user to clarify her request without appearing uncooperative (Wizard1 is better than just “The YP has no details of the quality of restaurants”) or stupid and stubborn (Wizard2 is better than “What type food/restaurant would you like?”).

User requests and system limits

The third case deals with the limits of the wizard’s knowledge. The user request has been understood, i.e. there is no need to clarify the portion of the database or limit the search with respect to some specification as in the previous examples, but the system cannot provide any information on the topic. These kinds of situations were common when the system had given a set of companies as the requested information and the user continued with follow-up questions.

- (13)
- | | |
|----------|--|
| User1: | are they all licenced to sell alcohol |
| Wizard1: | The YP cannot provide this information |
| | |
| User2: | which ones are open after midnight |
| Wizard2: | The YP cannot be specific; however, in general
such restaurants are open until 2 am |
| | |
| User3: | do any of them have a special student night |
| Wizard3: | The YP cannot provide this information;
please contact the restaurants concerned |

The adjacency pairs in (13) exemplify three different cases where the wizard has not been able to provide any information from the database, but attempts to compensate the failed evaluation by providing some related information. In the first case, the wizard has not been able to provide any other information than the statement of the limits of its knowledge. Notice that the simple answer “no” would have been untrue unless

the system knows about the particular restaurants' alcohol licence. In the second case, the wizard is able to offer some extra information which may be useful for the user. Although the YP database has not got any information on a particular restaurant, the world knowledge contains general information about the opening hours and this is included in the response. In the last case, the wizard offers a helpful rerouting instead of extra information. The user request has been associated with some information in the world model (e.g. that some restaurants have special student nights), but this is not general information about restaurants: it is not a general case that the restaurants have special students nights, but a special feature of individual restaurants. This causes a rerouting response: the user is asked to contact the restaurant to get the special information.

Another kind of compensation is exemplified in the dialogue (14) below. Now the wizard connects American food with American restaurants, and the user's request to describe the food is interpreted with respect to the task that the wizard is able to perform, namely to give YP-information about restaurants.

- (14) User1: describe what american food is like
 Wizard1: Sorry I do not know. Would you like details of the
 American restaurants?

The requests (13) and (14) can also be understood as questions where presuppositions (felicity conditions of the requests) are not fulfilled: the user believes that the wizard knows, or is able to do something, but this belief is not true. The wizard is thus obliged to correct the user's misconception, but also, if possible, provide helpful information or re-routing as compensation, related to its role as a helpful information provider: to show that the request has been understood and attended to in the most helpful way, relevant extra information is *actively* searched for and given to the user.

These examples can be compared to those that have been studied as presupposition failures, e.g. Kaplan (1983). However, our examples show the wizard's cooperativeness is not only restricted to pointing out and correcting false presuppositions, but it also includes active initiatives concerning compensatory information or re-routing. Consider, for instance, the following dialogue excerpt where the correction of the user's

misconception ”there exists a car hire company called ‘Salford car-hire’ ” is not found helpful enough, and it is accompanied by a compensation offering to give a list of car hire companies located in Salford.

- (15) User1: will Salford car-hire deliver?
 Wizard1: Sorry. There is no information on Salford car-hire.
 Would you like the list of car hire companies in Salford?

The wizard knows that some car hire companies deliver the car⁷, and so the presupposition concerns the existence of ‘Salford car-hire’ which has not been mentioned earlier in the dialogue. The wizard does not know any car hire companies with that name, but she appears to understand a common naming convention with companies: the proper name modifier often refers to the location of the company. Thus she is able to connect the request to her role as a YP information provider, and formulate a question that clarifies the user request in this regard.

5.1.5 Conclusion

If the collected dialogues and their application domain (Yellow Pages database) are considered from the ‘system design’ point of view, a traditional, system-oriented conversation mechanism specifically tailored for this and analogous situations is probably regarded as the most helpful and appropriate. However, the aim in this thesis is not just to improve usability of a single application, but to study the pragmatic principles behind the design decisions which are considered important in NL interface systems in general.

Written dialogues between a human and a (simulated) system have particular characteristics (e.g. lack of conventional politeness mechanisms, see discussion in 2.3.2), and it is not reasonable to aim at naturalness in the same sense as in human-human conversations. Even though a dialogue manager which could conduct a ‘real dialogue’ with the

⁷The immediately previous dialogue had, indeed, dealt with the delivery service of another car hire company.

user remains an idealised and not necessarily even a desirable goal, we consider communication management capability to be an important property of any human-computer interface, since the system's communicative capacity contributes significantly to the robustness of the system in general. On the one hand, it contributes to the usability and habitability of the system by allowing the user to express her requests conveniently. On the other hand, the design of a system can be based on general communicative principles and thus the portability of the system to other applications increases (cf. Ahrenberg et al., 1994).

In the sample dialogues, the fluency of communication is dependent on the inferences that the participants are able to draw, and on the adequate reporting of the result of the evaluation in the dialogue context. Specifically, if the wizard had provided responses like 'I don't understand what you mean' in (10), (11), left out the justification or repeated the same question in (12), or replied simply with "no, I do not know" in (13), (14) and (15), the fluency of communication would have suffered seriously and reinforced the impression of a stupid, uncooperative computer. The examples also show that the appropriate way to react in a situation where the request is vague or exceeds knowledge limits depends on the dialogue context (what has been talked about earlier), relevant information available (general and specific world and application knowledge), and the obligations that the role of the speaker sets for the communication (maximally cooperative and helpful information provider).

The extension of the dialogue system's robustness by implementing pragmatic principles also requires extensive knowledge about the world and the application domain. Content-related issues like metonymy and synonymy can be resolved with world knowledge reasoning. This brings robustness into connection with knowledge representation problems, widely acknowledged as one of the key current issues in AI research (Brachman et al., 1992). The construction of a world model that supports robust understanding was one of the aims in PLUS, but falls outside the scope of this thesis.

5.2 Desiderata for dialogue systems

In information seeking dialogues, the goal of the information seeker is to get information and the goal of the information provider is to give the requested information. A common strategy to deal with this kind of dialogues is **reactive** modelling, where the system responds to each user request separately, without reasoning about the dialogue context.⁸ However, it requires that the user knows what she wants and how she can obtain the information from the system. Even though the system may use some linguistic enhancements like ellipsis interpretation, the user must be somewhat familiar with the application and how to express the queries clearly and unambiguously.

Our empirical studies show that the users sometimes express their queries vaguely, and they also continue with follow-up questions (cf. Moore & Swartout, 1990). This requires that the system is able to act appropriately in situations which are not directly predictable from the previous utterance, but require contextual reasoning and monitoring of the user's beliefs and intentions. Moreover, the participants try to achieve their own goals by means of communicating about the assumed common purpose, and this requires that the system is able to initiate clarification of vague user contributions as well as actively provide extra information to help the user to reach her goal. By modelling the system as a rational cooperative agent, we also allow it to be persistent in its goal which contributes to the robustness of a system by means of cooperative conflict resolution. These requirements can be met by dialogue modelling which is based on general communicative principles and respects the negotiative nature of dialogues. We call this kind of dialogue modelling **constructive**.

In the constructive model, the starting point is the user with a need or desire which the system tries to satisfy as well as it can. The goal of the system is to construct a model of the 'object of desire'⁹ that the user query deals with, abstract the prototypical event within which this entity can be dealt with and then give appropriate information. Consequently, dialogue management is a way of controlling the achievement of this

⁸This is a strategy e.g. in library catalogue systems or French Minitel type query systems.

⁹This is a term informally used in PLUS discussions.

goal: it should ensure the optimal functioning of the communication as an instrument for achieving the purposes.

Below we discuss desiderata for constructive dialogue management in terms of CAA. The subsections deal with dialogue coherence, adequate system initiatives, joint purpose, communicative obligations, interactive communication, and conflict resolution. The examples are drawn from our dialogue corpus. Finally, we sketch the dialogue management on a conceptual level. Chapter 6 presents the system architecture.

5.2.1 Dialogue coherence

In Section 3.1.3 we discussed how dialogue coherence can be modelled either with the help of a dialogue grammar which defines acceptable and thus, coherent, structures, or with the help of the participants' goals and intentions, tied to the task structure. We do not use either of these models: we do not assume a pre-ordered structure with dialogue constituents, nor do we follow a task structure.¹⁰ Instead, coherence is captured by compliance with the obligations of communicative responsiveness, ideal cooperation and rational agenthood. We aim at *local* coherence, which holds between consecutive dialogue contributions. It is anchored in domain-based relations between discourse referents (see Chapter 7). As in Grosz et al. (1983), we associate each contribution with a single 'backward-looking center'. This is the Central Concept of the the contribution and it serves as the point of view from which the rest of the contribution is presented. *Global* coherence is attained by the participants' reasoning about appropriate – albeit not necessarily the best – responses on a local level, taking into account the dialogue context which has been created by the earlier contributions as well as 'persisting' in getting one's own goals satisfied (on persistence, see Section 5.2.3). It is not explicitly modelled in any global structure.

A dialogue consists of a sequence of contributions, each contribution being formed on the basis of the *updated dialogue context*. The dialogue context records the goals,

¹⁰Our task structure is so general that it does not give any support for coherence: the tasks are to get and provide information.

intentions, attitudes, discourse referents, Central Concept and NewInfo of each contribution. It has a flat structure, created incrementally as the dialogue goes on. Allwood and Haglund (1991) describe a sequence of utterances as “locally coherent” if the “incoming” obligations and options of a contribution match the “out-going” parts of the immediately preceding one. Loosely following Allwood and Haglund (1991), we can describe the dialogue progression schematically as in Figure 5.1.

Constructive dialogue management is in accordance with Bunt (1990, 1991) and Beun (1989) who formalised the idea of contributions containing context changing potential by defining communicative acts as functions from context to context. However, our approach is an elaboration of this in that we specify the context in a more detailed way (the mapping function is more complex) and relate the ‘context changing potential’ to wider communicative principles of cooperative and rational agenthood.

Constructive dialogue management can also be compared to the approaches by Suchman (1987) and Sibun (1991) who generate text on the basis of local context, without trees. Instead of hierarchically organised text structure, Sibun associates the coherence of texts with the domain structure of the subject matter. The text is thus sensitive to local constraints on what can be said next, modelled with the help of focus.

5.2.2 System initiatives

The capability to take initiatives and clarify vagueness is an essential factor in assessing dialogue participants’ conversational adequacy (cf. 5.1.4). Initiatives are, of course, dependent on the participants’ roles and mutual status in the activity they are engaged in. We have already mentioned in Section 4.6 that the system’s role as a simple information provider restricts its capability to introduce new topics and assist the user in her domain planning.

We define initiatives with respect to the achievement of communicative goals: an initiative is taken if at least one of the speaker’s goals is unfulfilled and needs to be pushed forward, or if the partner’s goal is vague and needs to be clarified. The system’s ini-

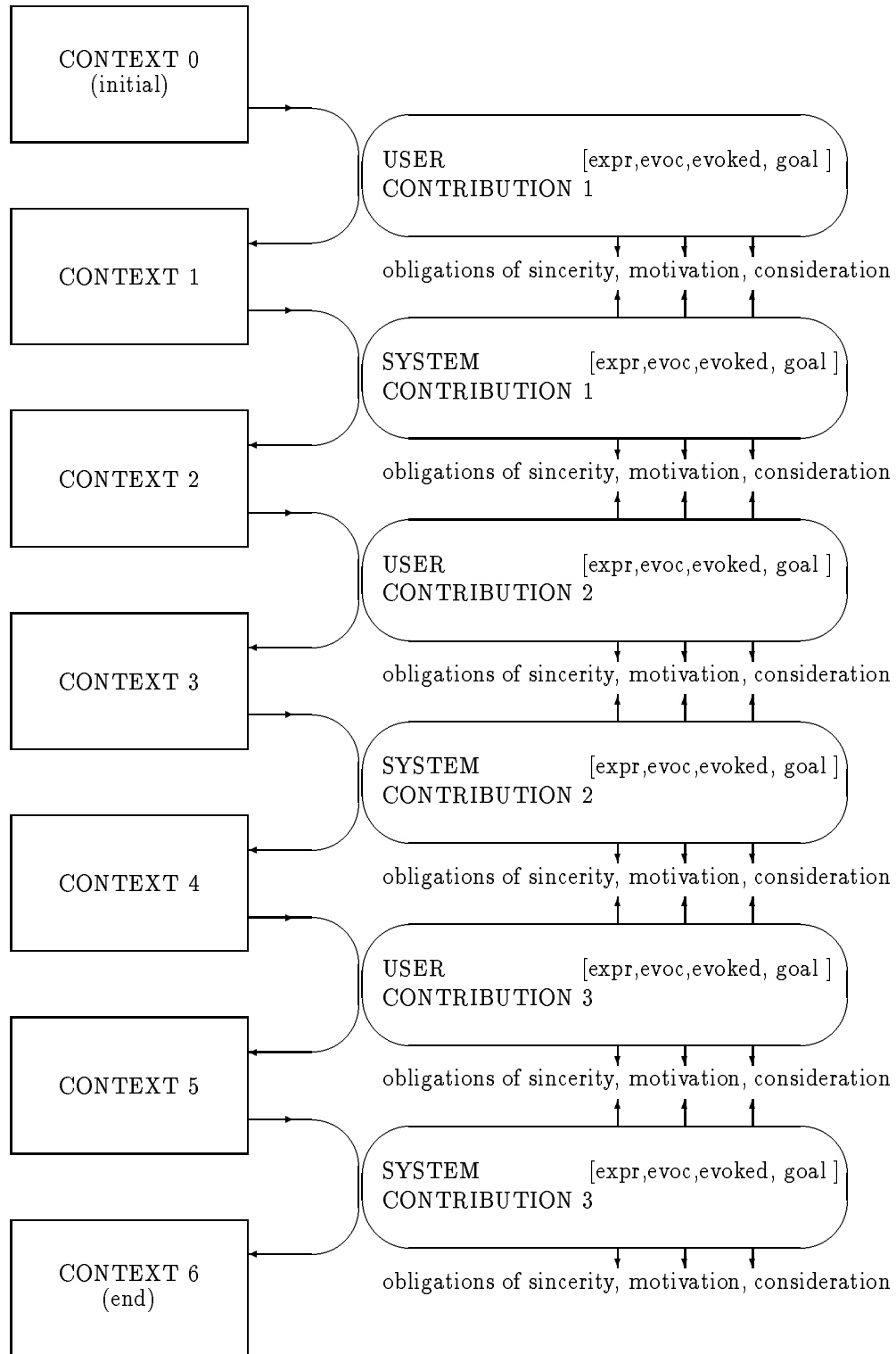


Figure 5.1: Relations between sequential contributions and dialogue contexts.

tiatives are restricted to goals that contribute to the system's role as a cooperative information provider and enable it to perform this role in a robust way.

In information-seeking dialogues studied in this thesis, the system can take the initiative if the user's request is:

1. too wide to start a database search,
2. too wide to give a reasonable set of database entries,
3. too vague to start a database search,
4. inconsistent with what the system knows about the world and the application,
5. inconsistent with what the system knows about the dialogue context,
6. not understood at all.

The first alternative appears in a situation where the user request has been mapped onto the application domain, but the mapping is ambiguous: there are several portions of the database that could be relevant for what the user requests. For instance, the PLUS dialogue (11) (p. 82) is of this type: the system needs to know whether to give the user information on car hire companies or car garages.

The second alternative refers to situations where the system has formulated a query to consult the application model, but the information associated with this query is too large, i.e. the number of entries exceed a certain limit which is considered appropriate to be given to the user at once. The first wizard contributions in the dialogues (10) (p. 81) and (12) (p. 83) are examples of this.

Even though the dialogue corpus does not contain good examples of the third case, we include it in the list in anticipation of cases where the system can interpret the user request with respect to its world model, but is unable to map the request onto any relevant information in its application domain. An example of this kind would be an opening *I am going to work in Bolton*, where the system has to specify what kind of YP professions, if any, can be associated with working in Bolton. Also, if the system did not know the relation between food and restaurants, the first user contribution in dialogue (10) (p. 81) would belong to this type.

The alternatives (4) and (5) occur in situations where the system's knowledge is limited or contradicts what the user says. The user request seems to contain misconceptions or false presuppositions about the system's knowledge (examples are found in dialogues (13), (14) and (15), p. 84 – 86), or there is a possible misunderstanding between the partners as in the following dialogue:

- (16)
- User1: Do they deliver the car?
Wizard1: Yes. They have a delivery service.
User2: is it for 25 hours?
Wizard2: you can rent a car at Mitchells self-drive for one day
or more and they also do hourly contracts.
User3: Yes, but is the delivery for 24 hours.
Wizard3: Sorry no such information on the delivery.
Usually the companies only deliver at office hours.

The last alternative (6) is for cases when the system cannot make any sense of the user request. The initiative depends on the communicative level on which the misunderstanding has taken place: e.g. if the system's 'perception' failed (parser could not parse the input), the system asks for rephrasing, or if the limits of the system's knowledge were encountered (e.g. YP information does not contain information about nuclear physics), the system informs the user about appropriate domain.

The system cannot take the initiative in dialogues except in these six cases. However, if its role were e.g. a cooperative advisor, the system could also take the initiative on the basis of what it knows about the task and the user's expertise level: the advisor may need to introduce topics not brought up by the user because the user is unaware of their importance for the goal.

Ideal cooperation requires that the participants trust each other to behave in accordance with the requirements of rational motivated action. Hence, the user is also assumed to behave cooperatively, and respond in a way that would not prevent the system from fulfilling its goals. Although the user can take initiatives, change topics and end the dialogue freely, she is obliged to give the requested information when the system asks for clarification or specification (cf. the discussion in Chapter 4.6.)

Webber and Joshi (1982) list three ways in which system can take the initiative in

constructing a response. As a negative answer to the question *Is John taking four courses?*, the system can

1. point out incorrect presuppositions:
No. John can't take any courses; he's not a student,
2. offer to monitor for the requested information:
No, three. Shall I let you know if he registers for a fourth?
3. justify 'why', i.e. provide grounds for the system's response:
No, three - CIS531, CIS679 and Linguistics 650.

In our classification these three cases can be seen as specifications of the alternatives (4) and (5): the system takes the initiative to clear up situations where the system's knowledge limits are encountered or the user request contradicts what the system knows. The responses are analogous to our examples (13), (14) and (15) (p. 84 – 86). The compensation offered in the second one, monitoring, can be derived from the system's role as a 'secretary' (this is not modelled in our information provider system).

5.2.3 Joint Purpose

A cooperative information provider tries to react in the way the user intended to evoke. However, if the system always adopts the user's evocative intentions, the system can only provide simple answers to clear and unambiguous user requests. As already mentioned, this is a simplified view of information seeking dialogues, which resemble negotiation processes rather than straightforward question-answer sequences.

An important and distinguishing feature in our system is that it is allowed to be persistent in its goals. Cohen and Levesque (1990a, 1990b) introduced the notion of a 'persistent goal', and defined it with respect to the speaker's rational activity. According to them, it is not rational to pursue a goal if the goal is already fulfilled or if the agent thinks that it is impossible to achieve it. Our notion of 'persistence' is related to what Cohen and Levesque advocate, although we specify it with respect to a wider notion of rational agenthood, discussed in Section 4.2. Simply, it is not rational to pursue all goals all the time, or some goals at all: persistence of a goal depends on the

agent's competence and the rational motivation for pursuing the goal. For instance, if the goal achievement turns out to be dangerous or unpleasant or there are more adequate and efficient ways to achieve the intended purpose, the goal is usually also dropped. The aspects of rationality subsumed by the requirements of the adequacy of the actions and pleasure and pain, are fixed in our system as a static aspect of dialogue management and not explicitly included in its reasoning.

Finding a joint purpose is the process that takes care of the evaluation of the user goal with respect to the system's own goals and determines if and how the system can continue the dialogue. We do not want to exclude the possibility that dialogue participants have their own goals which are not shared by their partners. However, communicators must have at least one common goal for the communication to succeed (transfer of information, cf. Section 4.3), and the process of finding a joint purpose encodes this.

There are four aspects, derived from the principle of ideal cooperation and rational agenthood, which we consider important in determining the joint purpose:

1. evocative intentions of the contribution (expectations),
2. evocative intentions of the previous contribution (initiative taken),
3. thematic relevance of the content of the contribution (Central Concept),
4. unfulfilled goals in the dialogue state (motivation).

Expectations

The conventional communicative function (illocutionary force) associated with a surface form provides conventional expectations of dialogue continuations: if the user has asked a question, the system tries to give an appropriate answer, but if the user has given an answer, the system tries to achieve its goals which are still unfulfilled. Of course, the speaker's response need not conform to the conventional expectations. The speaker may initiate a question after a question, give feedback or state something unrelated. In these cases the system needs to determine the joint purpose according to the principles of ideal cooperation, see examples below.

We do not advocate the ‘literal force hypothesis’ (see Section 3.2.1), since we do not classify communicative acts according to their functional types. Rather, we emphasise the inferential nature of communicative function on the basis of their propositional content and the dialogue context. Moreover, the act can be multifunctional, i.e. there may be several communicative functions that the act carries (see Section 6.1.2). Conventional communicative functions serve as the anchoring point for further analysis.

Initiatives

If the speaker has initiated a goal, we say that the speaker “has the initiative”. Then she also has the *right* to expect a response that fulfils the goal, and if the partner does not fulfil this expectation, she has the right to be persistent and pursue her goal until satisfied. In our system, initiatives are made by formulating a question with the evocative intention to elicit an answer, and the expected response is a statement with the evocative intention to make the partner believe the content of the statement. The system can take the initiative in the six cases discussed above.

Thematic coherence

Being able to produce coherent dialogues is one of the signs of rational cooperative agenthood: the speaker shows that she has understood what the partner said and is willing to collaborate in the joint task to achieve a mutually acceptable result. We model thematic coherence with the help of Central Concept which provides a view-point for presenting the new information of a contribution. Coherent shifts in the view-point are based on domain relations, see Chapter 7.

Of course, if the system had a more complex task structure, constraints in terms of related discourse referents would not be sufficient in determining the dialogue coherence. Instead, a reference to plans and/or rhetorical relations would be needed. These constraints can be included either in the domain-based topic shifting rules or in a separate task-model.

Unfulfilled goals

The communicative goals deal with the tasks of providing and eliciting information and they are stored in the expressive and evocative attitudes of the contributions. If the partner's response does not address the attitudes, the goal is stacked for future processing and a new goal taken up. The system stacks goals only if a brand new request is taken up, or if the user response is vague and needs clarifying. It reverts to an old goal when the expectations in the current dialogue situation are fulfilled and it is the system's turn to take the initiative. The unfulfilled goals form a simple LIFO stack, so they are satisfied in the reversed order in which they have been introduced. If a goal is already satisfied on the basis of the updated context, the next goal is tried. Thus the system need not revert through each goal explicitly, but it can jump over a goal if this is already satisfied. This is a further flexibility in comparison with dialogue grammars, cf. the discussion in 3.2.2. If all the stacked goals have been fulfilled in the course of the dialogue, the system concludes that the purpose of the dialogue has been achieved and initiates a question whether the user wishes to find some other information.

Conflicts can occur if the speaker has introduced a goal but the partner does not fulfil the expectations or replies with an unrelated response. Since the expectations and relatedness are derived from the assumption that the partner is cooperative, rational agent and will not attempt to prevent the speaker from fulfilling her goals, the speaker is faced with a conflict situation where she needs to decide on the appropriate way to proceed. She has to either find a motivation for the partner's non-compliance with the principles of ideal cooperation by proceeding with the new goal, or conclude that the partner has failed to process the previous goal in the intended way, and continue her previous goal, see Section 5.2.6.

Figure 5.2 presents different alternatives for the joint purpose depending on the four conditions in the context as defined above. The situations are assumed to be general, and thus the terms 'speaker' and 'hearer' are used instead of 'system' and 'user'. The examples refer to the sample dialogues given below in the explanation. Conflicts are marked with *X*.

case	initiative	expectation	topic	goals	joint purpose	example
1a	speaker	expected	related	unfulfilled	backto, clarify * ¹¹	17 – 20
1b	speaker	expected	related	fulfilled		
1b	speaker	expected	unrelated	unfulfilled	repeat-new, <i>X</i>	21
	speaker	expected	unrelated	fulfilled	* ¹¹	
2a	speaker	non-expected	related	unfulfilled	subquestion, <i>X</i>	22, 23
	speaker	non-expected	related	fulfilled	* ¹¹	
2b	speaker	non-expected	unrelated	unfulfilled	repeat-CC, <i>X</i>	24
	speaker	non-expected	unrelated	fulfilled	* ¹¹	
3a	partner	expected	related	unfulfilled	follow-up	25
3b	partner	expected	related	fulfilled	follow-up	26, 27
4a	partner	expected	unrelated	unfulfilled	new question	28, 29
4b	partner	expected	unrelated	fulfilled	new request	30, 31
5a	partner	non-expected	related	unfulfilled	backto	32, 33
5b	partner	non-expected	related	fulfilled	somethingelse	34, 35
6a	partner	non-expected	unrelated	unfulfilled	notrelated	36
6b	partner	non-expected	unrelated	fulfilled	new request	37, 38

Figure 5.2: Possible joint purposes for the system on the basis of contextual information.

Corpus examples

Below we explain each case with sample dialogues. In a few cases no examples were found in the corpus, and an invented dialogue is used as an example; the participants are then called ‘user’ and ‘system’. In the examples, the italicised wizard responses indicate the contributions planned when the constraints hold.

1. The **speaker** has the initiative, and the partner reacts with an **expected** response (statement). In our Context Model, the speaker has **unfulfilled** goals, since the initiative can only be taken if the partner’s request needs clarification or specification.

¹¹Situations where the speaker has the initiative and the partner has given an answer, but there are *no* unfulfilled speaker goals creates asymmetry between the partners in our Context Model. If the system has the initiative, there also exists at least one unfulfilled goal: to provide the user with some information. Thus these situations are impossible for the system. If the user has the initiative and the system has provided a requested list, the user may continue the dialogue with a new request or quit. Because we will consider the conditions from the system’s point of view, we have used the star ‘*’ to mark impossible system alternatives in the table.

- (20) User1: I need a car.
 Wizard1: Do you want to buy or rent one?
 User2: I need to go to the airport.
 Wizard2: *So, you want to rent the car?*

The wizard is, however, able to reason that the user probably wants to rent a car. She knows that going to the airport can be done either by car or by train, and because the user has expressed a need to have a car, she probably wants to go to the airport by car. Moreover, the wizard knows that buying a car means long or permanent use of the car, whereas renting a car is for temporary or time-bound use. The wizard also knows that going to the airport is a time-bound action, and thus the user apparently needs a car for a time-bound use, i.e. for renting. The wizard's goal, to know if the user wants to buy or rent a car, is thus satisfied. The conclusion has been drawn after a fairly long reasoning process, based on several assumptions, and the wizard wants to make sure that this is what the user wants, thus initiating a confirmation question.¹² The marker *so* here carries information about the long reasoning process which has resulted in the conclusion given in the rest of the contribution.

- (b) If the response is **unrelated**, the speaker repeats the initiative and also lets the partner know that the contribution was unrelated. See Section 5.2.6 for a more detailed discussion about this kind of conflict situation.

- (21) User1: I'm looking for a restaurant
 System1: Where?
 User2: My friend comes from Entwistle to visit me.
 System2: *Sorry I don't understand how this is relevant here. Where are you looking for a restaurant?*

2. The **speaker** has the initiative and the partner reacts with a **non-expected** response (question). As above, the speaker must have **unfulfilled** goals. These situations are considered conflict situations, see Section 5.2.6.

- (a) If the partner's response is **related** to the speaker's initiative, the speaker resolves the conflict by postponing her own goal and adopting the partner's

¹²The confirmation is asked in a declarative form (the so called 'declarative questions', Beun (1989)), and thus the wizard has actually produced an 'indirect speech act'.

goal (in structural terms this means that a ‘subdialogue’ starts). The rationale behind this is that the user may have a more general plan which she is executing, but which the system is unaware of, and if the user can pursue her own goal, she can also make the system’s goal redundant as a side-effect of revealing her plan. (Cf. dialogues (32) and (33) which show how such a ‘subdialogue’ is ended.)

In (22), the user ignores the wizard’s question about the location and continues with her own goal. The wizard stacks her goal concerning the departure location and continues with the user’s goal about car-hire companies in Bolton. The next user contribution indeed gives an indirect answer to the system’s question, see example (25).

- (22) User1: I need a car to go to the airport.
 Wizard1: Where would you start your journey from?
 User2: Are there any car-hire places in or near Bolton?
 Wizard2: *There are 15 car-hire firms in Bolton.*

In (23), the user ignores the wizard’s offer to specify the car hire company and insists on getting car hire firms in Entwistle until the wizard makes it clear that there are no car hire companies in Entwistle itself. The wizard replies to the user question but since the response is negative, she also offers compensation by means of repeating the specification. (Compensation also serves as a way to put forward the wizard’s own goal to give information about car hire companies.)

- (23) User1: can I have a list of car hire companies in the
 Entwistle area.
 Wizard1: Where is Entwistle?
 User2: Entwistle is outside Bolton.
 Wizard2: Altogether there are 15 car hire firms in Bolton.
 Do you want a list, or can you be more specific
 in your enquiry?
 User3: Are there any in Entwistle itself?
 Wizard3: *No, but your could be more specific in your
 requirements; for example, the YP can supply lists
 of those firms which offer unlimited mileage,
 delivery/collection, one way rentals,
 cheap local rates.*

- (b) If the partner’s question is **unrelated**, the conflict is resolved on the basis of the fact that the speaker has the initiative, i.e. she has the right to decide if the partner’s response fulfils her goals or not. The speaker is thus persistent

with her goal and repeats it, but also makes the topic explicit, thus forcing the partner to be explicit about the possible topic change. By formulating the contribution as a statement, the initiative is given to the partner: she has the ultimate right to decide the topic of the dialogue.

- (24) User1: I need a car.
 System1: Do you want to buy or rent one?
 User2: Do you know any restaurants?
 System3: *I thought we were talking about cars, and wanted to know if you want to buy or rent a car.*

3. The **partner** has the initiative and reacts with an **expected** response (question), **related** to the previous topic. In other words, the partner asks a follow-up question.

- (a) If the speaker has **unfulfilled** goals, she has two alternatives: either to pursue her own goals or respond to the partner's goal. Both alternatives are allowed to the user, but the system's role as a cooperative information provider restricts the system to behave in an expected way only. From the system's point of view the situation is not a conflict: because the user has the right to expect an answer, the system is obliged to continue with the user's initiative. The user thus continues the 'subdialogue' that she started sometime earlier with a follow-up request (cf. example (22)), and the system's goals may be fulfilled in the due course.

In (25), the wizard's goal is to know the departure place for the car rental, but this is overridden by the user's query about the existence of car-hire companies in Bolton, followed by the specification about the wanted company. The wizard adopts the user's goal and continues the 'subdialogue' accordingly.

- (25) User1: I need a car to go to the airport.
 Wizard1: Where would you start your journey from?
 User2: Are there any car-hire places in or near Bolton?
 Wizard2: There are 15 car-hire firms in Bolton.
 User3: which one of these would provide me with a
 delivery to my home, collection from the airport
 and be relatively cheap?
 Wizard3: *Is your home in Bolton?*

- (b) If the speaker has **no unfulfilled** goals, the partner's follow-up question is taken up directly. This case differs from the previous one in that the speaker knows that the original request has been fulfilled and the partner can continue the dialogue with a specification question of the given NewInfo (26) or with a related new request (27).

(26) User1: What is the address of Mitchells self-drive?

Wizard1: Please wait...
Mitchells Self-Drive
Dawes Street
Bolton
0204 394803

User2: Do they deliver the car?

Wizard2: *Yes. They have a delivery service.*

(27) User1: indian restaurants in the city centre

Wizard1: < list >

User2: try Chorlton

Wizard2: < *list* >

4. The **partner** has the initiative and reacts with an **expected** response (question), but the question is **unrelated** to the topic of the dialogue. The partner has apparently shifted the topic and started a new request.

- (a) If the speaker has **unfulfilled** goals, the speaker has two alternatives: either to pursue her own goals or take up the partner's goal. As above, the user is allowed to proceed in both ways, but the system can only continue with the user goal. The role again constrains the system's possibilities: it cannot initiate questions unless they are related to what has been talked about. Thus the first alternative cannot occur, and the user is never faced with questions that are unrelated to the topic.

The response differs from the follow-up responses (case 3 above) in that a remark about the unfulfilled goal is added. This causes the system to have the initiative and the user to state explicitly if a new request is started. In (28), the evaluation of the user request is negative and the remark about the unfulfilled goals also serves as a compensation. In (29), the user is asked to choose between two alternatives.

- (28) User2: What types of restaurants do you list?
 System2: Indian, Mexican, Chinese, Italian, Thai.
 User3: Which serve hot and spicy food?
 System3: Indian, Mexican and Thai restaurants serve
 hot and spicy food.
 User4: Do you know bus time tables?
 System4: *Sorry no. Do you want to continue with restaurants?*
- (29) User2: What types of restaurants do you list?
 System2: Indian, Mexican, Chinese, Italian, Thai.
 User3: Which serve hot and spicy food?
 System3: Indian, Mexican and Thai restaurants serve
 hot and spicy food.
 User4: What car hire companies are there in Bolton?
 System4: *There are 15 car hire companies. Do you want to
 see them or continue with the restaurants?*

These examples can be compared with the example (20) above. In case (1b) the speaker has the initiative which justifies her in being persistent with her unfulfilled goals, while here the partner has the initiative and the speaker can only make a ‘remark’. In both cases, however, the speaker controls the initiatives: in (1b) the speaker ‘keeps’ the initiative by being persistent, here the speaker ‘takes’ the initiative by requesting an explicit statement of the topic change.

- (b) If the speaker has **no unfulfilled** goals, a new request is assumed. This is the case e.g. at the beginning of the dialogue (the user request is unrelated to the system greeting, since the topic of the dialogue is determined by the user request). It can be compared with the dialogue (37).

- (30) Wizard0: Welcome to EYP. Please state your query.
 User1: How many car hire companies are there in Bolton?
 Wizard2: *There are 15 car hire companies.*

In (31) the wizard has given the initiative to the user by asking if she wants to continue the dialogue, and the user responds with a question. The answer to the wizard’s question is ‘no’, implicit in the actual expression of what the user wants to know. The wizard treats the user question as a new request which in this example is not understood at all.

- (31) Wizard10: Is that all you wanted to know?
 User11: Is bolton to rochdale to the airport o.k?
 Wizard11: *Sorry I do not know.*

5. The **partner** has the initiative, but reacts with a **non-expected** response (statement) which is **related** to the topic. The partner has commented on either the immediately previous contribution or one earlier on.

- (a) If the speaker has **unfulfilled** goals, she reverts to some previous unfulfilled goal. A ‘subdialogue’ started by the partner has obviously ended, and the partner has given a response to something that the speaker has initiated earlier.

In dialogue (32), the user has already asked about insurance consultants in South Manchester, and now wants to get information about car insurance. As a (non-expected) response to the wizard request in Wizard6, the user produces a counter question about the alternatives she can choose from, and then selects one of them. The wizard can now go back to the goal to give information about insurance agents. In order to fulfil this goal, the location of the service needs to be confirmed which causes the elliptical question.

- (32) User6: have you any information on car insurance.
 Wizard6: What type of insurance service do you want?
 User7: What type of services there are available?
 Wizard7: You can choose one of the following categories:
 Insurance administration services
 Insurance agents
 Insurance brokers
 Insurance companies
 Insurance consultants
 User8: Insurance agents please
 Wizard8: *Still in South Manchester?*

In the modified dialogue (33), the user has asked different restaurant types and then selects one of the alternatives presented by the system. The user also provides the location, and the system can revert all the way to the original goal and give the requested list of restaurants.

- (33) User1: I'm looking for a restaurant.
 System1: In which area?
 User2: What types of restaurants do you list?
 System2: Indian, Mexican, Chinese, Italian, Thai.
 User3: Try Indian in Rusholme.
 System3: *Please wait....*
 Here's the list <>

- (b) If **no unfulfilled** goals exist, the partner has obviously given evaluative feedback about the speaker contribution. The speaker can then ask if the partner wants to continue the dialogue, i.e. look for information on something else, or probably compensate by some other means. For instance, in (34) the user gives related feedback in two consecutive contributions: User2 is related to the driving-event and User3 to the general evaluation of the given information.

- (34) User1: which is the cheaper service
 Wizard1: If cost is important to you, you may be
 interested to know that self-drive car hire is
 cheaper than the chauffeur driven services.
 User2: i can't drive
 Wizard2: *You should contact the company for details of cost.*
 User3: this is not very helpful
 Wizard3: *Do you require further information from this service?*

In particular, we regard thanking as evaluative feedback, related to the previous topic. The dialogue (35) is an example of this.

- (35) User1: Can you tell me the best car-hire firms
 in Bolton and Manchester Airport?
 Wizard1: <list>
 User2: Thank you very much.
 Wizard2: *Is that all you wanted to know?*

It should be pointed out that the initiative is understood as belonging to the partner, even though the speaker contribution is formulated as a question. This is so, because the speaker contribution is understood as a check-question: the speaker wants to check if the partner wishes to continue the dialogue or not. Alternatively, it could be formulated as a statement (a declarative question). The interrogative mood is preferred since it puts more 'reactive pressure' on the user who can enter a new request as in (31), or end the dialogue explicitly as in (38).

6. The **partner** has the initiative and reacts with a **non-expected** response (statement) which is **unrelated** to the topic.

- (a) If the speaker has **unfulfilled** goals, the speaker lets the partner know that the contribution was unrelated and repeats the previous goal. This can be

compared to the situations in dialogues (28) and (29), where the system responds to the question but also reminds the user of the unfulfilled goals.

- (36) User2: What types of restaurants do you list?
 System2: Indian, Mexican, Chinese, Italian, Thai.
 User3: Which serve hot and spicy food?
 System3: Indian, Mexican and Thai restaurants serve hot
 and spicy food.
 User4: My friend comes from Entwistle to visit me.
 System4: *Sorry I do not understand how this is relevant
 here. Do you want to continue with restaurants?*

- (b) If the speaker has **no unfulfilled** goals, a new request is assumed. This is the case e.g. at the beginning of the dialogue, and it can be compared with the dialogue (30) where the user response is an expected question.

- (37) Wizard0: Welcome to EYP. Please state your query.
 User1: I need a car.
 Wizard1: *Do you want to buy or rent one?*

Dialogue (38) is analogous to dialogue (31) in that the wizard has asked the user to confirm further continuation of the dialogue. Now the partner's reaction is negative, and the dialogue is terminated. (If the user had replied *Yes. I need a car*, the situation would be like the example above.

- (38) Wizard10: Do you require further information from this service?
 User11: No thank you
 Wizard11: *Thank you for calling. Bye.*

Figure 5.3 summarises the joint purposes from the system point of view, and also provides a comparison for each alternative. Those marked with *X* are conflict situations.

5.2.4 Obligations

Obligations derive from rational agenthood and the principles of ideal cooperation. They are regarded as norms that rational, ideally cooperative agents follow, or, from the analysis point of view, as assumptions about the partner's rationality and cooperativeness. They are context sensitive, and thus what is obliged in one activity might not be obliged in another activity.

¹³This combination of contextual features is impossible, see footnote 11.

response	topic	goals	speaker initiative	partner initiative
expected	related	unfulfilled	backto, clarify	follow-up
		fulfilled	*13	follow-up
	unrelated	unfulfilled	repeat-new, X	new question
		fulfilled	*13	new request
non-expected	related	unfulfilled	subquestion, X	backto
		fulfilled	*13	somethingelse
	unrelated	unfulfilled	repeat-CC, X	notrelated
		fulfilled	*13	new request

Figure 5.3: Summary of the possible joint purposes for the system on the basis of contextual information.

1. **Sincerity** refers to truthfulness with regard to the current goals: the speaker is obliged to give information which she knows is true or for which she can provide evidence, and when asking questions, she is committed to a real need for the information.
2. **Motivation** deals with the question ‘why’. It relates the agent’s activity to her own strategies and normality and rationality conditions: questions are asked if there is a reason to elicit more information (to specify, clarify), and answers given if they contribute to the mutual understanding and assumed overall purpose of the dialogue.
3. **Consideration** encodes the agent’s rational and ethical consideration towards the partner. Basically, it means that the information is useful to the partner and does not prevent her from acting rationally. Misunderstandings and lack of understanding should thus be corrected, since they usually form obstacles later in the dialogue or in some other dialogues. The user is also entitled to get an explanation if the evaluation was negative (knowledge base limits were encountered, database search failed, Context Model became inconsistent, parsing was partial), or justification for why a particular request could not be carried out successfully. Answers should also appear in an appropriate format and amount, i.e. convey the intended message in an efficient way, free from false implicatures.

In PLUS, the obligative dimension was taken care of by two pragmatic rules: the *Obligation Fulfilment Rule* and the *Belief Transfer Rule*. The former says roughly that the reactive pressures put on the partner by a previous contribution must be subsumed by the current contribution, and the latter makes use of the sincerity of rational agents by assuming that the speakers have evidence for what they express and intend to evoke in the partner. That is, if an agent knows that the dialogue partner believes some proposition P and that the partner also has evidence for believing P , then the agent may herself believe that P is the case. In our system, the Obligation Fulfilment Rule is part of the process of finding the joint purpose, and we have also elaborated it to include conflicts. The Belief Transfer Rule is adopted as such. We also have a *Compensation Rule* which takes care of the different compensatory offers that the system can make, and several rules to encode the obligations of motivation and consideration. Examples of these are given in Section 9.5.3 together with their implementation.

The requirement of truthfulness, subsumed under the obligation of Sincerity, is also related to the ethical consideration, since a speaker may break the Sincerity obligation by not telling the whole truth, although still be obeying the ethics of being considerate, cooperative, or polite.¹⁴ This is also pointed out by Galliers (1989), who cites Goffman's (1970) notion of 'strategic interaction': everyday communication contains various grades and types of dishonesty, all of which should not necessarily be considered sinister.

5.2.5 Interactive communication management

Besides fulfilling the obligations of communicative cooperation, the system also has to manage interaction with the user. Allwood et al. (1992) divide interactive communication management into three types: turn management, sequencing and feedback.

¹⁴For instance, the speaker may not tell what she really thinks, because this would be embarrassing or considered rude, because she wants to please the hearer, or because the behavior is motivated by a want to save either one's own or the partner's face, see Goffman (1976), Brown and Levinson (1978).

Turn management deals with the means of assigning, accepting, taking, maintaining, and yielding the turn. In Conversation Analysis, turn management has been one of the main research topics (Sacks et al., 1974). In our system, turn-taking is enabled simply by the user hitting the return-key and the system signalling the end of its turn by a particular prompt. More elaborate methods could be used, but they fall outside of the scope of the thesis.¹⁵

Sequencing refers to the global arrangement of the communicative contributions with regard to subactivities and topics, i.e. to what has been discussed in terms of task planning or domain planning (Litman, 1985; Cawsey, 1993). In general, it commonly deals with initiation (opening, entering an activity, a subactivity or topic), maintenance (maintaining a subactivity or topic), changing (changing a subactivity or topic), and ending (closing an activity, subactivity or topic). In our system, these activities correspond to the general phases that are found in the collected dialogues: opening, requesting, follow-up questioning, and closing, and they are managed with the help of Central Concept and the system's knowledge about the achievement of the common purpose of exchanging information. The system need not be involved in all these activities as shown in our dialogue studies: the users commonly left out opening, follow-up questions and closing ceremonies.

Feedback refers to linguistic mechanisms by which the speakers can exchange information about the basic communicative requirements (contact, perception, understanding and attitudinal reactions), and is thus an important instrument in building joint understanding of the dialogue purpose. Allwood et al. (1992) present research on linguistic feedback in spoken interaction, especially its semantic and pragmatic features, and

¹⁵In PLUS, the collected French dialogues showed that the users often started another query while the wizard was looking for the answer to the original one. On the one hand, this points to the requirement that the system should have some sort of buffering mechanism and criteria to choose and change its reaction according to the queries in the buffer, i.e. cope with simultaneous turns and immediate feedback. On the other hand, one of the reasons for the user's simultaneous contributions was the slowness of the simulated system (the French Minitel was used as the application back-end), and this again points to the efficiency requirements: system responses should appear in a reasonable time so as to prevent the user from unnecessary contact disturbances.

show how feedback is important for the communication to flow relatively smoothly. Their point of view is that of a recipient rather than an initiator, and a typical feedback is given by heavily context sensitive single feedback particles like ‘ok’, ‘well’, ‘fine’ and ‘no’.

Their wide sense of ‘feedback’ seems to refer to exactly the same process as what we have called ‘determining joint purpose’, but there is an important difference which makes the topic of this thesis and the study of feedback in Allwood et al. (1992) complementary rather than opposite: although the speaker’s contribution provides regulatory information about how the joint purpose is being understood, the contribution is not just a feedback to what the partner said, but a *reaction* which actively controls the course of the dialogue. Our main concern is the planning of a meaningful response rather than the evaluation of linguistic feedback¹⁶, and in this we consider it important to distinguish the partner’s reaction in general (feedback in a wide sense) from the explicit use of pragmatic particles in a contribution (feedback in a narrow sense). It is also important to point out, following Allwood et al. (1992), that negative feedback on any level is always explicit, while positive feedback is usually implicit, and encoded in the speaker’s coherent and cooperative responses (cf. the dialogue 31 above). We therefore reserve the term ‘feedback’ for the narrow use only.

5.2.6 Conflicts

Galliers (1989, p. 199) notices that the common view of cooperativeness in the AI world, that agents are always in agreement and ready to adopt the other’s role, makes the systems rigid and unrealistic. If a system abstracts away from conflict situations so that they never occur because the context is constrained in a particular way, or if the system always adopts the partner’s goal simply because the partner has the goal, the system is benevolent rather than cooperative. Based on the work by Cohen and Levesque (1990a, 1990b), Galliers proposes a framework in which the agents can be

¹⁶Mainly because linguistic feedback in our sample dialogues was limited to a few pragmatic particles like ‘no’ (rejection, disagreement) and ‘ok’ (acceptance, agreement).

induced to “change their minds” so that conflict situations are resolved.

We will briefly discuss conflict resolution in Constructive Dialogue Management in this section, although it should be noticed that the chosen dialogue type, information-seeking dialogue, does not usually give rise to conflicting goals in the same sense as they are discussed e.g. in Galliers (1989). Moreover, to model the control that an agent has over decisions concerning which goal to pursue, some measurement of ‘preferences’ and the agent’s autonomy are required (cf. Galliers, 1989). These issues are beyond the scope of this thesis.

However, the system is capable of taking initiatives in order to clarify and specify vague user contributions and thus it can form its own goals which need to be satisfied with the user’s help. If the user response does not address the system’s evocative intentions (which have been formulated on the basis of the system’s cooperative evaluation of the user’s possible goal), a conflict situation may occur. The system must evaluate whether the user is still in contact with the system, whether she has perceived and understood the system contribution, and whether she is willing to continue the dialogue. Given that these basic communication requirements are ok, the conclusion is that the system goal conflicts with the goal assumed to be held by the user, and the system must decide on an appropriate way to proceed, and in particular, which goal to satisfy.

We have already in Section 5.2.3 singled out the four contextual factors which constrain the joint purpose. Potential conflict situations occur in cases where the speaker has own goals but the partner gives an unrelated and/or non-expected response. There are six such cases in our system, of which three result in a situation that we call a conflict and which require resolution on the basis of rational agenthood. Conflicts correspond to situations where the system has the initiative, since this is analogous to its having some control over the dialogue (the participant who has the initiative also has the right to pursue her goal unless this appears not to be rational). In the other three cases, the system does not have the initiative, and it will behave in accordance with the user’s evocative intentions, postponing its own goals if it has any. This reflects the system’s role as an obedient information provider: no conflict occurs if the system acts according to its role.

The dialogues (21) and (24), repeated here for convenience as (39) and (40), respectively, exemplify the two conflict situations where the system has the initiative and unfulfilled goals to pursue, but the user reacts with an unrelated contribution.

- (39) User1: I'm looking for a restaurant
 System1: Where?
 User2: My friend comes from Entwistle to visit me.
 System2: *Sorry I don't understand how this is relevant here. Where are you looking for a restaurant?*
- (40) User1: I need a car.
 System1: Do you want to buy or rent one?
 User2: Do you know any restaurants?
 System3: *I thought we were talking about cars, and wanted to know if you want to buy or rent a car.*

In example (39), the user has given an expected response (statement) and the system repeats its previous question. The system keeps the initiative, thus showing that it has understood the user's cooperative attempt to provide the appropriate information but that its capacity to relate the user contribution to what has been talked about is limited. The user is expected to give a comprehensible and related response, given that she behaves in a cooperative and rational way. In example (40), the user has given a non-expected response (question) and the system reminds the user of the topic of the dialogue as well as the question the user is expected to answer. The system gives the initiative to the user, and in this way actually acknowledges the user's right to manage the dialogue by initiating new topics and ignoring the previous half-finished requests. The system's persistence is persistent only to the extent that it does not conflict with the system's role as a humble information provider.

The third conflict situation occurs if the user reacts in a related but non-expected way to a system initiative. The user contribution signals that the user is in contact and has perceived the previous dialogue, but either she has not understood the system's question or is unwilling to continue the direction initiated by the system. In this case, the system lets the user 'manage' the dialogue: it stacks its own goal and adopts the user's goal instead. As mentioned in the joint purpose case 2a (p. 101), the rationale behind this is that the user may have a more general plan which she is executing, but

which the system is unaware of, and by letting the user pursue her own goal, she may also make the system's goal redundant as a side-effect of revealing her plan. It may also be that the system has misunderstood the user's goal, and if the system follows its own interpretation, it may appear uncooperative and stubborn. If the system has correctly understood the user's goal, it can always return to the question later, given that the user has not provided the missing information otherwise. In example (33), repeated as (41), the user indeed gives the missing information about the location together with the restaurant's type in her contribution User3, and thus satisfies the postponed system goal explicitly, while in example (42)¹⁷, the system needs to go back to its previous unfulfilled goal and explicitly ask the same question.

- (41) User1: I'm looking for a restaurant.
 System1: In which area?
 User2: What types of restaurants do you list?
 System2: *Indian, Mexican, Chinese, Italian, Thai.*
 User3: Try Indian in Rusholme.
 System3: Please wait....
 Here's the list <>
- (42) User1: I need a hire car.
 System1: Chauffeur-driven or self-drive?
 User2: Well, I just need to get from my office to the airport.
 System2: *Where is your office?*
 User3: In the City Centre.
 System3: Is it just a one-way hire?
 User4: Yes.
 System4: And do you want a chauffeur-driven or self-drive car?
 User5: Chauffeur-driven.
 System5: Ok. Here's the list <>

If the user has the initiative but does not continue in a thematically related way, and if the system has unfulfilled goals, the situation is again problematic. However, we do not consider these cases conflicts but rather 'puzzles', where the system needs to find an appropriate continuation although the user has not provided it with consistent clues about what is the 'object of desire'. The system assumes that a new request has started, but reminds the user about the unfulfilled goals. These cases are exemplified by examples (28) and (36), given here as (43) and (44), respectively.

¹⁷A PLUS dialogue, invented by Jean Carletta.

- (43) User2: What types of restaurants do you list?
 System2: Indian, Mexican, Chinese, Italian, Thai.
 User3: Which serve hot and spicy food?
 System3: Indian, Mexican and Thai restaurants serve hot
 and spicy food.
 User4: Do you know bus time tables?
 System4: *Sorry no. Do you want to continue with restaurants?*
- (44) User2: What types of restaurants do you list?
 System2: Indian, Mexican, Chinese, Italian, Thai.
 User3: Which serve hot and spicy food?
 System3: Indian, Mexican and Thai restaurants serve hot
 and spicy food.
 User4: My friend comes from Entwistle to visit me.
 System4: *Sorry I do not understand.*
 Do you want to continue with restaurants?

5.3 Conclusion

We have abstracted four characteristics for dialogue systems from the corpus of information-seeking dialogues: physical feasibility of the interface, efficiency of reasoning components, ability to cope with syntactically problematic input, and ability to initiate responses that clarify vagueness or misunderstandings, and we have discussed how they are related to the CAA principles of rational motivated activity. Especially, we have concentrated on the participants' initiatives to clear up vague or misunderstood contributions, and claimed that this is a distinctive feature in improving dialogue systems' robustness. We have considered factual information-seeking dialogues, where the system's role is to provide truthful and helpful responses and not to take part in more complex planning. However, it is assumed that the communicative principles apply to other types of dialogues, too, where the system's role is more dominant and it has more initiative as regard to topic introduction, the user's decisions and planning.

We also described how 'Constructive Dialogue Management' can be used to model the negotiative nature of the dialogues. The Allwoodian concept of ideal cooperation (participants have a joint purpose, show cognitive and ethical consideration, and they trust each other to act according to these constraints) is encoded in the six dimensions of constructive dialogue management: coherence of dialogues, capability to take initia-

tives, determination of joint purpose, obeying communicative obligations, interactive communication, and conflict resolution.

No particular dialogue structure grammar is needed, since global coherence is guaranteed through general communicative principles. As discussed in Section 3.2.2, the problem with any structurally-oriented approach is that it requires the predefined structures to be used in a standard way, and we believe that Constructive Dialogue Management brings the desired flexibility to analysis and planning processes. At the same time it also meets theoretical and empirical requirements about the negotiative nature of dialogues.

We do not classify utterances according to speech acts either. As discussed in 3.2.1, the current work on speech act theory (Cohen & Levesque, 1990a, 1990b; Perrault, 1987; Beun, 1989; Sperber & Wilson, 1987; Levinson, 1983) considers the recognition of speech act type unnecessary. Instead of classifying and generating communicative acts according to a taxonomy of acts with preconditions and effects, we base them on a theory of rational agenthood and cooperation.

Chapter 6

Design of a Constructive Dialogue Manager

In the previous chapter, we discussed the four desiderata for a robust dialogue system, and dwelled especially on the system's ability to initiate responses that clarify vagueness or misunderstandings. We developed the 'Constructive Dialogue Management' approach to dialogue management on the basis of CAA principles of ideal cooperation and rational agenthood. This approach considers the ability to deal with the following dimensions to be essential for a robust dialogue system:

- coherence of dialogues,
- initiatives,
- joint purpose,
- communicative obligations,
- interactive communication,
- conflict resolution.

In this chapter, we study how these desiderata can be met in the system design. We first give an overview of the dialogue manager architecture and present the units and concepts used in the system. We then sketch how Constructive Dialogue Management

works and discuss some key issues in response planning. Finally, we study how cooperativeness is addressed in response planning and compare the approach to the relevant literature.

Preliminary work for the research described in this chapter was carried out within the PLUS project, but the main results reported here derive from independent research. The system presented in this thesis is original, but benefitted from the ideas and general objectives of the PLUS system. The first prototype of the PLUS system is described in Bego et al. (1993), and the PLUS response planner is discussed in Jokinen (1993b).

6.1 Dialogue manager architecture

6.1.1 Overview

Our dialogue system can be schematically depicted as in Figure 6.1.

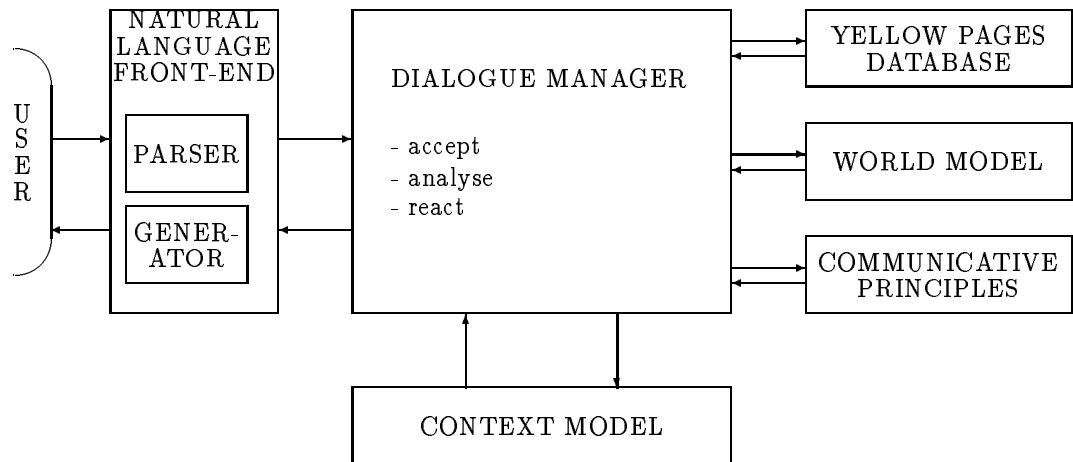


Figure 6.1: The architecture of the system.

The user is connected to the system via a terminal. A shallow natural language front-end mediates the queries to the dialogue manager. The dialogue manager is the controller of the main level reasoning processes and it has three tasks: accept an input,

analyse the input and react to the input. Acceptance of a user contribution deals with the evaluation of content and perception, i.e. reading the user input and parsing it. Analysis contains two subtasks: determining the user goal and NewInfo, and determining Central Concept of the contribution. It is part of the system whose detailed implementation falls out of the chosen scope of the thesis. Reaction has two subtasks: formulation of the system's own goal and planning of a response. Both tasks are reasoning processes effectively using contextual knowledge sources.

The key resource of the Dialogue Manager is the Context Model. This is a dynamic knowledge base containing information about contributions, discourse referents, a distinguished discourse referent Central Concept, new information, goals, and expressive, evocative and evoked attitudes. The concepts are defined in Section 6.1.3, and the corresponding data structures are described in Section 9.1.

The dialogue manager also has access to three static knowledge bases: Yellow pages database (application back-end), world model knowledge base, and the communicative principles. In our system, the world model is a small knowledge base designed for the purposes of the research presented in this thesis. It is based on the PLUS World Model described in Cavalli et al. (1992b), but differs from it in that it uses a simplified representation of concepts and contains rules which are tailored to make distinctions considered important in this work. It contains knowledge about the application domain as its subpart. Application back-end is a small database consisting of car hire companies and restaurants in the Manchester area. This database, used in the empirical work, was compiled by Daniel Jones and is described in Jokinen et al. (1991).

Communicative principles refer to the obligations of rational, cooperative communication, and they are encoded as a set of inference rules dealing with the knowledge in Context Model. Part of the system's communicative knowledge is encoded procedurally in the control algorithm of the task 'reaction': e.g. the constraints of ideal cooperation (the communicators should have a joint purpose, they should obey communicative principles and show cognitive and ethical consideration in their contributions, and they should trust that the partner behaves in a rational way) are separate steps in the specification of the system response from the goal to a surface contribution.

The detailed description of the implementation is given in Chapter 9.

6.1.2 Context Model

Dialogue management deals with information of seven different types:

1. conventional, context independent meaning of the contributions (parser output and surface generator input),
2. new information and Central Concept of the contribution,
3. expressive and evocative attitudes carried by the user and the system contributions,
4. dialogue history (a record of the dialogue as a sequence of contributions),
5. world knowledge,
6. application knowledge (Yellow Pages database),
7. communicative principles.

As mentioned above, a novel feature in our system is that it does not use a dialogue structure grammar to structure the dialogue. Instead, information is encoded in the Context Model as Prolog facts, and the dialogue structure emerges incrementally as the dialogue proceeds. The Context Model is a dynamic knowledge base and it contains information as shown in Figure 6.2. The data structures are explained in detail in Section 9.1.

For the purposes of testing the response planner alone, the previous dialogue context can also be specified by the predicate `context/9`. The first argument is the context number to identify the context, the second one is the speaker, the third one the communicative goal, and the fourth one the contribution as a string of words. The fifth, the sixth and the seventh arguments encode expressive, evocative and evoked attitudes, respectively. The eighth argument is Topic and the ninth NewInfo. A list of instantiated discourse referents is encoded in a separate predicate `discrefs/2` which encodes the contribution number and the discourse referents available.

Contribution	<code>contr(ContrNumber,Speaker,Sentence).</code>
Goal	<code>goal(ContrNumber,Speaker,GoalStates).</code>
Unfulfilled goals	<code>unfulfilledGoals(ContrNumber,UnfulfilledGoal).</code>
Expressive attitudes	<code>ex(ContrNumber,Attitude).</code>
Evocative attitudes	<code>ec(ContrNumber,Attitude).</code>
Evoked attitudes	<code>ev(ContrNumber,Attitude).</code>
Discourse referents	<code>dr(Concept).</code>
Central Concept	<code>topic(ContrNumber,CentralConcept).</code>
NewInfo	<code>new(ContrNumber,NewInfo).</code>
Gathered information	<code>infoGathered(Query,Specs).</code>

Figure 6.2: Information in Contextual Knowledge Base.

An example of a context represented this way is given in Figure 6.3. This context is assumed to prevail after the first user contribution *I need a car* at the beginning of the dialogue, before the planning of the system response starts. The evoked attitude corresponds to the system goal after the evaluation of the user goal, filtered with respect to the application model. On the basis of the world model, the system has inferred that the user can have a car either by buying, renting or stealing, and it also knows that the way in which it is able to help the user in this goal is that it supplies information on appropriate Yellow Pages services. However, as the Yellow Pages database has no information about stealing cars, the system concludes that the user intends to either buy or rent a car.

6.1.3 Concepts and units

We will use the following concepts and units in describing dialogue organisation. Those marked ‘PLUS’ are directly based on the definitions used in PLUS.

Contribution: (PLUS)

This is the basic communicative unit of dialogue, and it serves as an anchoring point for other organisational units such as the right to communicate (turn taking), grammatical

```

context(1,user,[want(u, know(s,[user(u), needEvent(n,u,c), car(c)]))],
/* contribution: */
['I need a car'],
/* expressive attitudes
explicit: */
[want(u, know(s,[user(u), needEvent(n,u,c), car(c)]))],
/* implicit: */
[want(u, know(s,[wantEvent(w,u,h), haveEvent(h,u,c), user(u), car(c)]))],
/* evocative attitudes
explicit: */
[want(u, want(s, know(s,[user(u), needEvent(n,u,c), car(c)])))]],
/* implicit: */
[want(u, want(s, know(s,
[wantEvent(w,u,h), haveEvent(h,u,c), user(u), car(c)])))]],
/* evoked attitudes: */
[know(s,[user(u), needEvent(n,u,c), car(c)]),
know(s,[wantEvent(w,u,h), haveEvent(h,u,c), user(u), car(c)]),
/* before AM filtering:
know(s, know(u,
[wantEvent(w,u,d), disj(d,b,r,s), buyEvent(b,u,c,_),
hireEvent(r,u,c,_), stealEvent(s,u,c,_), user(u), car(c)])),*/
know(s, know(u,
[wantEvent(w,u,d), car(c), user(u), hireEvent(r,u,c,_),
buyEvent(b,u,c,_), disj(d,b,r,s)]))],
/* topic: */
needEvent(n,u,c),
/* new: */
[user(u), needEvent(n,u,c), car(c)]
).

```

Figure 6.3: Representation of the context after the first user contribution
I need a car.

units, and functional intention units (goals, expressive, evocative attitudes etc.). A contribution is defined as a sequence of communicative behaviour bounded by lack of activity. One should notice that silence can be a very effective contribution and carry a lot of implicit information: "I'm embarrassed", "I don't know", "I don't want to talk to you", "Let's not talk about this", etc. Silence thus does not lack communicative activity, although it lacks physical activity. The unit in spoken language corresponding to a contribution is an *utterance*.

Turn: (PLUS)

This is defined as a right to communicate and is a normative rather than a behavioural unit like contribution. A turn is equivalent to an utterance in our system, since the

change of turn is indicated by mechanic prompts by the user and the system. However, as the following examples show, this need not always be the case in human-human conversations (from Allwood, 1992):

(45) A: It's raining?
B: Mm

(46) A: Don't go there
B: <silence>

In (45), B's utterance 'Mm' is simultaneous to A's turn, while in (46), B fills her turn with silence. Thus, an utterance need not be a turn, as in (45), and a turn need not be an utterance, as in 46. Notice that silence as such is not an utterance, although it can be a non-spoken contribution.

Grammatical units: (PLUS)

A contribution contains several grammatical units, such as sentences, phrases, and words. In particular, a contribution need not contain a full sentence, but an ellipsis, even a single word can function as the preferred form. In example (47) below, the contribution contains a pragmatic particle 'Sorry', a noun phrase ellipsis, and an imperative mood sentence. Linguistic structure is marked with punctuation marks and capitals.

(47) Sorry. No information available on prices. Please contact the company.

Communicative goal:

We define a goal as a set of related facts about the world which describe the desired state of the world. A communicative goal is a set of epistemic facts that describe the mental states of the partners: what the speaker knows or believes that the partner knows. For instance, a communicative goal might be that the epistemic fact

`know(s,[WantEvent(w,u,h),hireEvent(h,u,c),user(u),car(c)])`

(the system knows that the user wants to hire a car) is true in the context. A communicative goal is tied to the speaker's domain-related goals, such as 'hire a car', 'know about restaurants', or 'plan a trip', but in this thesis we only consider domain goals of the type 'get information on X'.

The goal is assumed to be a consistent set of facts. This is in accordance with Cohen and Levesque (1990a, 1990b), although it is a restriction concerning agents in the actual world: one may hold conflicting desires (have a cake and eat it).

The definition allows us to distinguish between desired states which the agent believes can be attained and the states which the agent believes are impossible to attain. For instance, the state of the user knowing the opening hours of a car hire company may be a desired state, but it is impossible to attain since the system has not got any information about the opening hours. Moreover, it allows us to draw a distinction between what the agent will plan to satisfy and what she, for some reason, is less committed to satisfy. For instance, the goal may contain a desired state that the user knows about the cardinality of the set of car hire companies, but on the basis of contextual information the obligation of consideration may discard this as unnecessary from the final message.

The two distinctions between goals (possible or impossible in the context and goals to which the agent is committed or not committed) follow the distinctions made in Cohen and Levesque (1990a, 1990b). Cohen and Levesque define a committed or persistent goal as one which the agent will give up only when achieved, or if the agent believes it is impossible to achieve, or if the reason for the goal is no longer true.

Communicative intention:

The communicative *intention* of the system is an intention to act so that the goal will be true. Allwood (1976, p. 13) makes a distinction between intentions that are connected to actions (e.g. an agent may intentionally connect to the electronic YP) and intentions that are connected to mental states (intention to get a car in the future), although his discussion mainly concerns the first type. Intentions vary in their degree of consciousness, and it is not necessary that prior to the performance of an action there is

full and clear awareness of the intentions (as in habits). Rather, there must be a certain disposition to the conscious awareness of the direction and purpose of the action. The system's communicative intentions are determined on the basis of the reasoning about the joint purpose which thus serves as the seed for what the system intends to say, and directs its behaviour towards a certain end. Depending on the context, the goal is specified either by augmenting it with other desirable and necessary states or dropping off some desirable states which are considered irrelevant. The system's joint purposes are listed in Figure 6.4.

Related to basic requirements	
quit	system wants to quit
beAble	system wants to be able to perceive (parse) the user contribution
toUnderstand	system wants to understand what the user said
Related to communication management	
toCorrect	system wants to correct a user misconception
somethingElse	system wants to know if the user wants information on something else (feedback)
notrelated	system has not been able to relate the user contribution in the dialogue context
repeatNew	system wants to repeat its previous goal
repeatCC	system wants to repeat its previous CC
backto	system wants to revert to a previous goal
followup	system wants to follow up the user request
newquestion	system wants to give the information and remind of CC
newrequest	system wants to provide information
subquestion	system wants to follow 'subdialogue'
clarify	system wants to clarify the given NewInfo
specify	system wants to specify information concerning one single database heading
toomanyheadings	system wants to clarify the scope of the database search

Figure 6.4: Joint purposes.

Communicative act: (PLUS)

Allwood (1976, p. 21) defines 'action' as a "certain type of behaviour which is governed by an intention in order to achieve certain desired effects". The effects are the purposes

of the action. On these lines we define ‘communicative acts’ as actions which are governed by a communicative intention to achieve a particular communicative goal, i.e. a desired state in the discourse world.

Communicative acts are also called ‘dialogue acts’ as in SUNDIAL (Bilange, 1992) and TENDUM (Bunt et al., 1984), to distinguish them from other acts that the system is capable of performing (assisting in planning, searching for information, giving expert information), and also to emphasise their difference from speech acts, which do not encode dialogue information as do dialogue acts.

Communicative acts are functional units of a dialogue. We want to emphasise, however, that our system will not classify communicative acts into communicative act types. Functional labels such as ‘question’, ‘request’, ‘statement’¹, ‘check’, ‘confirm’, etc. are regarded as abbreviations for certain combinations of contextual factors (the agent’s intentions, related attitudes, and communicative obligations), but they do not have any particular theoretical status in the framework. This is a view held commonly nowadays, cf. Cohen and Levesque (1985), Galliers (1989). Instead, communicative acts are characterised in terms of more abstract types of communicative functions: expressive and evocative. For instance, asking a question means expressing a desire for information and trying to evoke the desired information, and making a statement means expressing belief and trying to evoke a shared belief, cf. Allwood (1976).

Figure 6.5 shows how grammatical units, communicative acts and contributions relate to each other. It is based on the distinctions made in PLUS (Jokinen et al., 1992), but differs from it in that it lacks the level of dialogue structure units (Moves). The example consists of two contributions by the speakers *S* and *U*. In the first contribution, and in most cases indeed, the notions ‘contribution’, ‘communicative act’, and ‘grammatical unit’ coincide. In the second contribution, there are three communicative acts, two of which are included in the same sentence and realised as consecutive clauses. The first communicative act is realised as a separate grammatical unit. The mispuncting *tohire* divides the second communicative act into three grammatical units, recognised by the

¹Allwood (1992) considers the term “inform” as a synonym of “statement” misleading, since according to him, every contribution informs in some way or another.

parser as separate units.

S: Do you want insurance companies in Chorlton?

```
|-----GrU0-----|
|-----CommAct0-----|
|-----Contribution0-----|
```

U: No. I want to hire a car, because I'm going to see my old aunt.

```
| -GrU1- | -GrU2- | -GrU3- | -GrU4- | -----GrU5-----|
| -CAct1- | -----CommAct2----- | -----CommAct3-----|
| -----Contribution1-----|
```

Figure 6.5: The relative 'scopes' of grammatical units, communicative acts and contributions.

Communicative function: (PLUS)

CAA characterises communicative functions by:

- speaker intentions, i.e. the goals that they are supposed to achieve;
- the agent's related beliefs and other attitudes;
- the communicative pressures (obligations) which are put on the addressee and to which the agent commits herself.

Grammatical units carry communicative functions and the communicative function is part of a grammatical unit's contextual meaning. A contribution may consist of several grammatical units, but have a single communicative function, as B's response in the following dialogue which could take place in the middle of noisy road construction:

- (48) A: Shall we have a break?
B: Cannot hear you properly, but yes let's have a break.

On the other hand, a contribution and a grammatical unit can also be multifunctional: they can realise several functions simultaneously. For example, the contribution in example (47) above contains sequentially the functions: apology ('Sorry'), answer ('No

information available on prices'), and request ('Please contact the company'). Furthermore, the answer could simultaneously serve as an explanation and the request as a compensation.

Multifunctionality is not only connected to the main message, but it can also appear with respect to own communication management and interactive communication management, see Allwood et al. (1992). In these cases, the successive contributions contain traces of repair structures, sequences, turntaking and feedback. For instance, in the following contribution

(49) yes, I'm looking for an Indian eh Italian restaurant.

'yes' is a feedback word which is related to interactive management, 'Indian eh Italian' shows own communication management, and the main message is 'I'm looking for an Italian restaurant'. Since we are dealing with written dialogues, we do not pay attention to own communication management (although it is important when studying the planning of the spoken responses).

It is important to distinguish communicative functions from grammatical mood. Grammatical mood signifies the combination of a particular syntactic and morphological structure² and the common moods are:

- declarative
- interrogative
- imperative
- exclamative³

Grammatical moods are associated with a conventional function (statement, question, request and exclamation, respectively). However, as discussed in Section 3.2.1, a linguistic expression can also convey other functions not conventionally associated with

²See also Levinson (1983, p. 243), who points out that the term 'mood' refers to the categories of verbal inflection in traditional grammars ('imperative' contrasting with 'indicative' and 'subjunctive'), rather than to the use of sentences or utterance types.

³Allwood includes this in the moods, following the suggestion by Quirk et al. (1972).

its mood. For instance, a declarative sentence can be used as a question (called ‘declarative questions’ by Beun (1989)), and an interrogative sentence can be used as request (*Can’t you switch off that radio*).

Expressive and evocative attitudes:

The expression of speaker beliefs, desires, etc. is called the *expressive* function of a communicative act, and the intended evocation of beliefs and other attitudes in the addressee is called the *evocative* function of the act. The attitudes corresponding to these functions are called *expressive* and *evocative* attitudes, respectively, and they encode the communicative functions of the speaker contribution. For instance, if the speaker intends to inform the hearer about P , she *expresses* her intention to make P known to the hearer, while she *evokes* her intention to make the hearer to accept and believe P . Similarly, if the speaker intends to get information about Q , she expresses her intention to know about Q , and she evokes her intention to make the hearer make Q known to her. By dividing the attitudes associated with each communicative act into expressive and evocative attitudes, we have a basic mechanism to deal with the communication of the beliefs and other attitudes that the speaker holds herself, and those that she intends to evoke in the addressee.

Figure 6.6 shows the expressive and evocative attitudes associated with the three main grammatical moods⁴. ‘Sp’ refers to the speaker, ‘He’ to the hearer and ‘P’ is the content of the attitude. The table can be compared to Table 4.2 in Section 4.7.

Expressive attitudes	Evocative attitudes	Grammatical mood
want(Sp, know(He, P))	want(Sp, want(He, know(He, P)))	declarative
want(Sp, know(Sp, P))	want(Sp, want(He, know(Sp, P)))	interrogative
want(Sp, do(He, P))	want(Sp, want(He, do(He, P)))	imperative

Figure 6.6: Conventional association of expressive and evocative attitudes with the three main grammatical moods.

We also define *evoked* attitudes as part of the addressee’s reactions to the speaker’s

⁴We do not deal with exclamative sentences such as *A new car!*, as these are not very likely to occur in the information seeking dialogues.

contribution. The speaker assumes them to be evoked by the contribution for it to be successful, and thus they can be compared to presuppositions as well as felicity conditions of a successful communicative act. When the addressee is the system, they are evoked on the basis of inferences using the Context and world model.

The set of attitudes associated with a contribution represents the communicative act performed by the speaker by producing the particular contribution, although it should be noticed that there is no fixed collection of attitudes that makes a particular act. Rather, there are as many communicative acts as contributions performed, and the fact that they can be labelled as ‘inform’, ‘question’, ‘check’, and ‘confirm’ is due to their appearance in a particular functional position in the dialogue. However, the same collection of communicative attitudes can appear in different functional positions, and conversely, the same functional position can be occupied by different collections of communicative attitudes. The conventional association of expressive and evocative attitudes with the grammatical mood is used in generation so as not to complicate the task; thus e.g. the system’s intention to know something is always realised by a sentence with interrogative mood.

Both expressive and evocative attitudes can be explicit or implicit, the distinction being that explicit attitudes have (on the surface level) only explicitly expressed concepts as their content, while implicit attitudes also contain implicitly expressed concepts. Explicit attitudes are derived from the surface form of the utterance, while implicit attitudes are inferred on the basis of explicit attitudes, the system’s cooperative principle and the context. The set of expressive attitudes is not necessarily shared by both interlocutors, but we make this simplifying assumption. The set of evocative attitudes is the set of attitudes which are intended to be evoked in the hearer, and the nature of human-computer interaction requires that the user’s evocative attitudes are the system’s interpretation of what the user wants to evoke in the system. They are the system’s beliefs about the intentions of the user, inferred on the basis of the Context Model and world model.

Figures 6.7, 6.8 and 6.9 show examples of CAA-based analysis of the attitudes. The content of the attitudes (in [square brackets]) is expressed in world-model concepts

using Davidsonian event variables. The letters *u* and *s* refer to the user and the system, respectively, and the predicates *know* and *want* are belief and intention operators, respectively.

<p>EXPRESSIVE ATTITUDES OF THE INPUT:</p> <p>want(u, know(s, [user(u), needEvent(n, u, c), car(c)])) want(u, know(s, [wantEvent(w, u, h), haveEvent(h, u, c), user(u), car(c)]))</p> <p>EVOCATIVE ATTITUDES OF THE INPUT:</p> <p>want(u, want(s, know(s, [user(u), needEvent(n, u, c), car(c)]))), want(u, want(s, know(s, [wantEvent(w, u, h), haveEvent(h, u, c), user(u), car(c)])))</p> <p>EVOKED ATTITUDES FOR THE RESPONSE:</p> <p>know(s, [user(u), needEvent(n, u, c), car(c)]), know(s, [wantEvent(w, u, h), haveEvent(h, u, c), user(u), car(c)]), know(s, know(u, [wantEvent(w, u, d), car(c), user(u), hireEvent(r, u, c, -), buyEvent(b, u, c, -), disj(d, b, r)]))</p>
--

Figure 6.7: Expressive, evocative and evoked attitudes of the user contribution *I need a car*.

In Figure 6.7, the fact that ‘needing a car’ means ‘wanting to have a car’ is inferred on the basis of the world model. The attitudes with this content are implicit. Evocative attitudes comprise the assumed user intention, which is shared by the system due to the ‘Belief Transfer Rule’ (see Section 5.2.4): the system believes that the user needs a car and wants to have a car, since, being a rational agent, the user would not have expressed this if she had lacked evidence. A new system goal (the system wants to know if the user wants to buy or rent a car) is evoked after reasoning about how the system can fulfil the shared goal with respect to its application model: a want to have a car can be fulfilled by giving information about the companies which provide this kind of service. The system knows that buying or renting enables the user to have a car, and since it also has information about companies which enable both types of services, it needs clarify the user’s goal to know which service the user has in mind. The system knows that the user is an expert on her goals and knows which service she wants.

In Figure 6.8, the utterance is understood as a direct answer to the question of where the

EXPRESSIVE ATTITUDES OF THE INPUT:

want(u, know(s, [location(r, entwistle), hireEvent(r, u, c, _), user(u), car(c)])))

EVOCATIVE ATTITUDES OF THE INPUT:

want(u, want(s, know(s, [location(r, entwistle), hireEvent(r, u, c, _), user(u), car(c)]))))

EVOKED ATTITUDES FOR THE RESPONSE:

know(s, [location(r, entwistle), hireEvent(r, u, c, _), user(u), car(c)])

know(s, know(u, [hireEvent(r, u, c, _), location(r, Loc)]))

Figure 6.8: Expressive, evocative and evoked attitudes of the user contribution *In Entwistle*.

user wants to hire a car. The expressive attitude functions as the only user intention, and this intention is shared by the system. However, “Entwistle” is an unknown placename to the system and thus the system cannot properly evaluate the information given by the user. The system cannot continue with its original task (which was to identify a known placename where the user wants to hire a car), and clarification is formulated as a new system goal to know where Entwistle is located (see the joint purpose in example (18), p. 99).

EXPRESSIVE ATTITUDES OF THE INPUT:

want(u, know(u, [existq(co), carHireCompany(co), location(co, bolton)]))

EVOCATIVE ATTITUDES OF THE INPUT:

want(u, want(s, know(u, [existq(co), carHireCompany(co), location(co, bolton)])))

EVOKED ATTITUDES FOR THE RESPONSE:

know(s, [existq(co), carHireCompany(co), location(co, bolton)])

know(s, [cardinality(setOf(co, cos), 12), carHireCompany(co), location(co, bolton)])

know(s, know(u, showall(12, carHireCompany(co))))

Figure 6.9: Expressive, evocative and evoked attitudes of the user contribution *Are there any car hire companies in Bolton?*

In Figure 6.9, the user has asked if there are any car hire companies in Bolton. The expressive and evocative attitudes contain the expressed content of the existence of the car hire companies, while the evoked attitudes comprise not only this but also the cardinality of the companies. The attitude with the cardinality content comes from

the system evaluating the user input which results in the check of the database if any car hire companies are listed there. Thus the cardinality of the car hire companies comes as a side effect of the system trying to evaluate the user query in the first place. (Compare Joshi et al. (1984) who favour of a similar method.)

Discourse referents:

These are world model concepts instantiated in the Context Model. All events, i.e. activities, states and processes are represented in the Davidsonian first order logic with appropriate indices. They can thus be referred to by the index and considered discourse referents.

We also adopt the policy of Discourse Representation Theory (Kamp and Reyle (1993)) and assume that indefinite noun phrases introduce discourse referents. For instance, the user utterance *I need a car* instantiates three different discourse referents: the state of needing, the user and the car⁵:

`needEvent(n,u,c), user(u), car(c)`

Some information the speaker regards as more important, and this is foregrounded by linguistic and pragmatic means. Allwood (1978) calls this aspect of communicative status the ‘prominence of information’. However, he only considers the distinction between foreground and background information, which often coincides with, but is not always equal to, the distinction between new and old information. We make both distinctions: *aboutness* distinguishing foreground (topical) and background (salient) information, and *newness* distinguishing new (focal) and old information, see Section 7.1.2. We call the distinguished discourse referent which is being talked about the *Central Concept (CC)* of the contribution. It sets the common ground, or the viewpoint on what can be coherently talked about. In the earlier development of the system

⁵‘car’ has been introduced as a discourse referent, although it occurs as an object of an intensional predicate ‘need’ and should thus get a different status from extensional discourse referents. We do not go into details of the distinction between intensional and extensional predicates or discourse objects here.

CC was referred to as ‘topic’, and we sometimes use this term as well. In referring to the new information that is focussed on, we use the term *NewInfo*, not ‘focus’, to avoid terminological misunderstandings.

Aboutness and newness divide discourse referents into three different types, presented in Figure 6.10. ‘Topics’ are discourse referents which have been talked about, i.e. they appear as CCs and thus belong to old information. ‘Background’ contains those discourse referents that do not appear as CC, but have been explicitly introduced in the dialogue and thus belong to old information. ‘Salient’ discourse referents are those which are accessible by virtue of being thematically related to some CC, but have not been explicitly introduced in the dialogue. Thus they belong to new information in the dialogue context. The case were a discourse referent is CC and belongs to new information is impossible: new information can be made CC later in the dialogue but not in the contribution in which it is introduced.

Information Status	Aboutness	
	Central Concept	non-CC
NewInfo	impossible	Salient
OldInfo	Topic	Background

Figure 6.10: The different types of discourse referents.

6.2 Sketch of Constructive Dialogue Management

Constructive Dialogue Management equals the agent’s evaluation of the partner’s contribution on the four communicative levels (see Section 4.5): contact, perception, understanding and attitudinal reactions. The evaluation process is based on the contextual information, and it is governed by the principles of ideal cooperation: the agent trusts that the contribution has been formulated according to the requirements of joint purpose as well as cognitive and ethical considerations. Furthermore, the result of the evaluation must be communicated to the partner: communicative responsiveness obliges the agent to report the result to the partner. The reaction can be based on

any communication level: the speaker may report either positive or negative results on contact (*Go on/I should leave*), perception (*I can see the writing and I know the words/Sorry, the screen is blurred, there is a typo*), understanding (*I got the point/I don't understand*), and attitudinal reactions (*Ok, I agree/No, I disagree*). As pointed out in Allwood et al. (1992), if the evaluation of a contribution is negative on some level, explicit feedback is given to the user about the failing level, whereas if the evaluation is positive, the feedback is usually implicit: an adequate response shows in itself that the speaker is willing to be in contact, is able to perceive contributions, has understood the partner's contribution and reacted to in an appropriate way.

Of the four basic communicative requirements, the first two, contact and perception, are usually fulfilled by the very use of the system and by a non-distorted readable screen. We have used the status of the parser output as a way of implementing the system's 'perception' (Section 9.3). In a dialogue system that deals with spoken language, the perception of contributions becomes a more serious problem and requires special design decisions (e.g. the dialogue manager must make extensive use of check questions to guarantee that the contributions are perceived and analysed correctly, see Bilange (1992)). As mentioned in Section 5.1, efficiency of reasoning is also important in maintaining contact.

The requirement of understanding means that the dialogue partners should try to infer the relevance of the contributions in the context. They should see the rational motivation behind the contributions, in other words reason about the goal that the contribution serves to achieve. It is, of course, unreasonable to require the user to see the system as a rational, motivated agent, but at least it is feasible to assume that the user understands the motivation of the system responses with regard to the information-seeking task that the system is trying to perform. This can be compared to the requirement of 'transparency', "the system's capabilities and limitations should be evident to the user from experience" (Ahrenberg et al., 1994). The system, for its part, trusts the user to behave according to the principles of ideal cooperation. User contributions are interpreted as if they have a rational motivation and a meaningful connection with the dialogue situation, and thus the interpretation process establishes

a new dialogue context in which the system can start to plan its reaction.

The system's reaction is also based on the principles of ideal cooperation, but now the application of the requirements is 'the other way round': the system plans the contribution in such a way that the joint purpose is conveyed and the response obeys the communicative principles of Constructive Dialogue Management. Since the context has been updated by the previous contribution, the reasoning results in the system goal which is a set of desirable facts about the next dialogue state which the system wants to make true. The 'message' is formed from the goal by refining it with respect to communicative obligations, and encoded in the expressive and evocative attitudes that the contribution must communicate either explicitly or implicitly. In our system, the goal concerns the epistemic situation of the partners, i.e. what the agents know or want the partner to know. Next section discusses the generation task and presents the planning algorithms.

6.3 Pragmatics-based planning

6.3.1 The generation task

The system presented in the thesis exemplifies the 'slippery slope' of defining the starting point of generation, discussed by McDonald (1993). McDonald points out that the answer to the question 'how far back does generation go?' is tied to the proportional amounts of linguistic and contextual information in the specification which serves as the source of generation. He further advocates a 'no-expression, specification-less' view of generation: the message level on which one reasons about the information content of the utterance does not express the information as a 'message' in non-linguistic terms, but rather, the utterance is gradually specified in the course of the generation process, and the intermediate representations exist only to the point that is needed for the construction of the next level of representation. In our implementation (see Chapter 9), we have actually drawn this borderline in even more radical terms: the embryo of the system response is determined in the evaluation of the user contribution as the

result of the analysis of joint purpose. This goal is further specified with respect to the system's application knowledge, communicative obligations and surface realisation criteria in the course of planning the response.

The central problem in generation in general is choice (McDonald (1983), Dale et al. (1990), Hovy (1988a), McKeown and Swartout (1988)). In a similar vein, we identify the tasks of our response planner as follows:

1. determine the message and the 'communicative act' in terms of expressive and evocative attitudes,
2. address dialogue coherence by selecting the Central Concept and marking possible topic shifts appropriately,
3. determine the level of explicitness in a response (generation of ellipses),
4. select and disambiguate semantic predicates that realise the content of the goal,
5. choose referring expressions (pronouns, definite descriptions) on the basis of topic information.

The thesis concentrates especially on tasks (1) – (3),⁶ showing how the principles of CAA can be encoded in a working dialogue system, and effectively used in improving the robustness and adequacy of system responses. Chapter 7 discusses coherence and Central Concept determination and Chapter 8 deals with explicitness and ellipsis.

6.3.2 Planning algorithm

The planning starts from the assumption that it is uncooperative to express nothing: the obligation of responsiveness requires the report of the result of the evaluation of the partner's goal.

The planning algorithm can be verbally presented as follows:

⁶Tasks (4) – (5) are discussed in the PLUS deliverable by Bego et al. (1993).

1. After the system goal has been selected by the reasoning about the joint purpose, the communicative intentions are specified by referring to the obligations and commitments. The result is an ‘obliged’ goal which accords with the communicative and contextual requirements.
2. The ‘obliged’ goal is refined by determining the Central Concept which anchors the content of the goal into a thematically coherent view-point from which the goal can be presented. Central Concept is determined by predicting possible candidates in the dialogue situation on the basis of domain-based coherence relations and then selecting one according to communicative principles.
3. A semantic qlf-representation is planned for the ‘obliged’ goal. The planner uses two data structures: a stack which contains the explicit concepts that are to be communicated to the user (Agenda) and a list which collects the semantic predicates and forms the augmented semantic representation of the contribution (SemRepr). Initially the Agenda contains NewInfo only, but concepts can be pushed on to the Agenda according to the relevance algorithm (see below).
4. The qlf-representation is translated into a string of word forms by the Surface Generator.

The algorithm for relevance checking goes as follows:

1. Initialise the Agenda by pushing NewInfo into it as well as concepts marked ‘explicit’ (e.g. a CC that needs to be explicitly present in the surface contribution).
2. Check if the Agenda fulfills the four Relevance Criteria. If so, go to step 3. If not, add goal concepts to the Agenda according to the criteria which failed. The criteria are checked in the order: Accuracy, Consistency, Validity and Free From False Implicatures. If there are no goal concepts to be added, fail: the goal cannot be realised in an accurate, consistent and valid way which is free from false implicatures. Replanning must occur.
3. Translate the Agenda into a surface level representation via the Conceptual Lexicon. If there is no mapping from the concept to a semantic predicate, try to decompose the concept, and find a way to paraphrase the concept with concepts that are realisable. If this is impossible, fail: the goal cannot be realised on the surface level.

The relevance criteria of Accuracy, Consistency, Validity and FFI are tested one at a time, and their order represents increasing cooperativeness. If the speaker wants to convey truthfully her own goals only, it is necessary that the contribution is accurate. If the speaker wants to produce a contribution which also ‘makes sense’ with regard to her own goals, the contribution must be consistent. If the speaker wants to pay

attention to the dialogue context and to the partner, the contribution must be valid, or show how the contribution is connected to the previous discourse. Finally, if the speaker does not want to lead the hearer astray, the contribution must be free from false implicatures. The relevance criteria are discussed in Section 8.2.2.

6.4 Cooperative system responses

In this section, we show how communicative principles can be applied to the three cases of cooperation discussed in Section 3.3.2. Our approach is close to the work by Joshi et al. (1984), but we differ from this in that we seek for overarching principles of human rational activity, and aim to present how these can provide a unified basis for cooperative behaviour, and also for dialogue management and response planning in general.

We do not model cooperativeness by rules which explicitly encode Gricean maxims in terms of attitudes, but rather, it emerges from the global system design as the system attempts to fulfil the communicative obligations of sincerity, motivation and consideration, and to maintain consistency of the context. Since the system is cooperative, it wants to carry out the tasks implied by the user's evocative intentions and attempts to react in a way that accords not only with the system's own general goals, but also with what it has inferred to be the intended effects of the user contribution. From the system's view-point, cooperativeness is thus encoded in the requirement that the evocative dimension of the user contribution and the evoked system response must match: the system responds in a way that the user intended to evoke (however, cf. the conflict situations discussed in Section 5.2.6).

6.4.1 Cooperativeness and presuppositions

If the user's query reveals misconceptions or presuppositions which the system knows are false, the obligation of Consideration requires that the system clarifies these. For

instance, in dialogue (50), the evocative dimension of the user question concerns the number of car hire companies at Bolton airport.

- (50) U1: How many car hire companies are there at Bolton airport?
S1: None. There is no airport at Bolton.

The system simulates a motivated information provider and thus wants to offer the user information that would address the evocative dimension of the user request, i.e. inform the user of the number of car hire companies at Bolton airport. However, the evaluation of the user request results in a negative response, since the system knows that Bolton has not got an airport at all. It is inconsistent with respect to the system's knowledge base to assume that Bolton has an airport. According to the obligation of Sincerity, the negative result is given to the user, and according to the obligation of Consideration, the user is entitled to get an explanation of why her sincere, motivated and considerate request was not successful. The system thus formulates the message so that the user knows that there are no car-hire companies at Bolton airport and that there is no airport at Bolton. Contrary to Kaplan (1983), the presupposition is not corrected because the system wants to correct the user's false presupposition, but because it would not otherwise show that it has understood the user request and reacted to the expressive dimension of the utterance.

Another example of the system's 'over-answering' is the following sample dialogue:

- (51) U2: Are there any car hire companies in Bolton?
S2: Yes. 15.

As in HAM-ANS, cf. example (8), Section 3.3.2 above, the system provides the user with extra information which is not overtly requested by the question. In our analysis, however, the additional information results from the system's rational, motivated activity, not from the system's linguistic knowledge about appropriate case role fillers or more specific quantifiers. The evocative dimension of the user request deals with the existence of car hire companies in Bolton, and as before, the system adopts the user's evocative intentions. The existence of car hire companies is resolved by checking if

there are any instances of car hire companies in Bolton that the system knows of. The database search returns the number of car hire companies as well as the information about these companies. From this potentially relevant information pool (which actually is achieved as a result of the process of joint purpose), the system excludes the YP-information on the basis of the obligation of Motivation: since there are no user intentions in the context directly concerning this information (the user has expressed her interest in the mere existence of the companies, not in the list of actual companies). The cardinality of the found set is motivated, however, because it implicitly contains information about the existence of the companies: being non-nil, there are some car hire companies in Bolton. The location is not given, since it is in the immediate context, and moreover, the repetition of the location would cause the hearer to infer that there is some contrast between Bolton and some other place. The system also needs to comply with the obligation of Consideration, and the simple cardinality of the car hire company set as a response would be inadequate again: the system response has to show that the expressive and evocative dimensions of the user contribution has both been addressed, and thus the affirmative marker “yes” is added as a direct answer to the surface question about the existence of car hire companies.

6.4.2 Cooperativeness and plans

The system may give additional information together with a positive evaluation of a request if the motivation behind the request is understood. In certain dialogue situations such as (9) in Section 3.3.2, we may think of the motivation as a well defined task structure. The user was given the departure location as well as departure time, since not knowing the gate where the particular train leaves may be an additional obstacle to carrying out the plan. Similarly, if the user has asked the address of a car hire company, the system may give the user the telephone number and opening hours as well, to enable her to contact the company at the right time. However, this kind of cooperativeness requires that the additional information is highly relevant to the information directly asked, i.e. the relation between them is obvious. If the user is planning a one-way rental and has asked which car hire companies have branches both

at the city centre and at the airport, the information about the opening hours may not be relevant at all, and if given, confuse the user rather than be cooperative.

The relevance of the additional information is constrained by the obligations of Motivation and Consideration. A piece of information, e.g. the opening hours of a company, is given if it contributes to the overall goal of the dialogue, e.g. the goal of getting and providing information about car hire companies in general, but not if it is irrelevant to the goal, e.g. getting and providing information about which car hire companies provide one-way rentals. Such decisions can be effected with the help of the Central Concept: if the new piece of information is coherently related to the Central Concept of the contribution, it can be given. We can assume for instance that ‘opening hours’ is an attribute of the concept ‘company’, in the same way as ‘address’ and ‘telephone number’, while ‘one-way rental’ is an attribute of ‘hire event’. When the system is talking about ‘car hire company’, and new information concerns the properties of a given company, all the attribute values associated with the company, either as its particular properties or inherited from a super-concept, are forwarded to the user. The obligation of Consideration requires the system to give all the relevant information, so that the user need not ask for particular pieces of information separately.⁷ The information about the opening hours of a company is thus given together with its address and telephone number to the user. If the topic of the dialogue has been ‘hire event’ and the new information asked concerns the companies which enable one-way hiring, the response would start from a different set of relevant information: the car hire companies and not necessarily their properties are relevant.

6.4.3 Cooperativeness and false implicatures

Consider the following example:

⁷Remember that in corpus studies, milking of the information was considered one of the drawbacks of the dialogues.

- (52) User: Are there any car hire companies in Bolton?
S1a: Budget Rent-a-Car
Bradford House 287-289 Manchester Road
Bolton
0204 391611

Hertz
Manchester North branch
Bolton
061-273 8884
S1b: Yes, for instance:
Budget Rent-a-Car
Bradford House 287-289 Manchester Road
Bolton
0204 391611

Hertz
Manchester North branch
Bolton
061-273 8884
S1c: There are 15 listed.

The request evokes an answer to an existential query on car hire companies, and the system, being cooperative and sincere, checks if there are any instances of car hire companies in Bolton. If there are only two car hire companies in Bolton, the system will give their details as a response, following the obligations of Sincerity and Motivation. From the elliptical response, the user can conclude that the two companies listed in S1a are the only car hire companies in Bolton: had the system known of other companies, it would have given their names as well.

However, if there are other car hire companies in Bolton as well, the reply S1a in (52) is uncooperative, since it leads the user to believe that *Budget Rent-a-Car* and *Hertz* are the only car hire companies in Bolton. The system does not obey the obligation of Sincerity: it gives information that is not true with respect to its knowledge base. Moreover, the response violates the obligation of Consideration, since the information provided by the system is not useful for the partner: it permits the user to draw false conclusions about the number of car hire companies in Bolton.

If the user request is motivated by the intention to know if there exist some car hire companies in Bolton at all (e.g. the user wants to know which places in general have got

car hire companies and the previous context contains similar questions about car hire companies in different places in succession), the system can address the user's expressive and evocative intentions by the response S1b. This conforms to the obligation of Sincerity as well as Consideration: the system does not mislead the user to believe that *Budget Rent-a-Car* and *Hertz* are the only car hire companies in Bolton, but it also gives the exact amount of information that is requested by the user (the existence of car hire companies in Bolton) as well as justifies this response by giving two examples. Of course, there is a pervasive problem in such a system in that its database is almost certainly incomplete, and so S1b may always be appropriate. We thus prefer the response in S1c, which makes the number of car hire companies explicit and also conveys an implicature that the number is based on the companies listed in the system's database.

6.5 Conclusion

In this chapter we discussed Constructive Dialogue Management from the architectural point of view, defined the necessary units and concepts for Constructive Dialogue Management, and discussed how cooperativeness can be addressed in response planning.

By extensively using the dynamic knowledge in the Context Model and the static knowledge about the world, application and communicative principles, the dialogue manager decides on the appropriate joint purpose and its specification into a surface level contribution.

The general requirements for the system's planning of a contribution are that the response be:

- intentionally clear, i.e. show what is its communicative goal,
- contextually consistent, i.e. address the expressive and evocative attitudes of the previous contribution,
- thematically coherent, i.e. relate to some previously agreed Central Concept.

The planning adheres to the gradual specification of the system goal, discussed by McDonald (1993). Our system will first determine the joint purpose for the dialogue, which corresponds to the system goal formulation. However, this process only collects the information that is to be communicated to the user either explicitly or implicitly, and the contextually relevant content of the system response is specified by later planning processes which refer to communicative principles. The planner interleaves the two components: ‘what to say’ and ‘how to say’ and thus overcomes boundary problems between planning and realisation.

The planning of the propositional content of a system response starts from the new information. This is augmented with contextual information, if the (elliptical) contribution consisting of NewInfo only does not convey the goal appropriately.

Cooperativeness is something that is created simultaneously by the dialogue partners: since both partners are obliged to evaluate the contributions with respect to their own goals and report on the result of this evaluation, cooperativeness emerges from this mutual activity as a gradual but successful fulfilling of the goals set for the dialogue. We can also get a straightforward definition of uncooperative behaviour: if an agent cannot reach her communicative goals, or appreciate that they are being approached in the dialogue, given that the partner can perceive the contributions and is treated as a rational motivated agent, the partner’s behaviour is described as uncooperative, lacking the basic willingness to communicate and react in an appropriate way.

Chapter 7

Dialogue Coherence

In this chapter, we study how dialogue coherence is related to the overall framework of CAA. As discussed in Section 3.1.3, coherence is understood as consisting of inferable links between ideas, objects and events referred to in consecutive contributions. The linking of central entities (events or objects participating in the events) is based on domain or world model relations: for instance, when talking about a hire-event, possible coherent next topics include the hire object, the hire location and the reason for the hiring. However, in line with Hobbs (1979), we also emphasise that coherence is a deeper notion than “discourse just being about some set of entities”. While Hobbs assumes that the speakers’ desire to be understood drives them to seek for a suitable coherence relation between two sentences, we assume that the driving force is the communicative competence of the partners, as they analyse and evaluate contributions according to the communicative obligations in a dialogue context.¹ Dialogue coherence in terms of identifiable links between discourse referents is then an expression of the fact that the higher level communicative obligations have been appropriately addressed.

Levinson (1983, p. 315) emphasises that topical coherence is not based on shared reference across utterances, but rather “constructed across turns by the collaboration of

¹This does not exclude the possibility of implementing communicative coherence via Hobbs-type abduction. What we are claiming is that the abductive ‘explanations’ are based on rational and motivated considerations by the participants, rather than *ad hoc*-type coherence relations.

participants”. The overall topic of a dialogue is agreed as the conversation goes on, and new topics can be introduced by some chance association to the content of the previous turn (“Hey that reminds me of...”). Levinson argues in favour of research which concentrates on how potential topics are introduced and agreed, avoided and closed, collaboratively. However, he does not really address the question why this activity makes conversations coherent or why people communicate in such a way that the resulting dialogue is coherent. We attempt to show that coherence arises from the speakers’ rational and motivated behaviour through fulfilling communicative obligations. Introducing new topics, agreeing to discuss some topic, and closing a topic are decisions that the speakers make in order to attain the goal set for the dialogue. The decisions are governed by the speakers’ communicative competence: the evaluation of the partner’s contribution in the context results in a new joint purpose that needs to be conveyed to the partner, and the view-point taken to present this information must be chosen so that the contribution forms a coherent continuation with respect to the previous context.

The dialogue context contains discourse referents, and these provide the primed context within which contributions are analysed and produced. The priming effects discussed in Section 3.1.3 are thus described with the help of the discourse referents that have been introduced so far: what has been talked about earlier directs and constraints the interpretation of the coming information.

We study coherence from the point of view of generation and concentrate on the speaker’s task of ensuring that the response addresses the communicative principles thereby producing coherent dialogues. In Section 7.1 we first clarify the terms ‘topic’ and ‘focus’, and introduce our own terminology: ‘Central Concept’ and ‘NewInfo’. In Section 7.2, we discuss Central Concept with respect to rational agenthood and introduce topic shifting rules, formulated after McCoy and Cheng (1990), and study some corpus dialogues from the point of view of topic shifting. In Section 7.3 we present an example of how these concepts are used in our system.

7.1 Terminological distinctions

In text linguistics (Halliday & Hasan, 1976), the distinction between *cohesion* and *coherence* has been made, the former referring to surface level means which tie the text together (e.g. the use of pronouns to refer to previously mentioned discourse referents), and the latter to semantic linkings between sentences. The distinction is useful as a descriptive means to distinguish overtly marked from covertly inferred, but when it comes to analysing or producing coherent text or dialogue, communicative purposes of the speakers must be attended to: one also needs to reason about the information content being conveyed. For instance, pronominalisation indicates that one of the discourse referents has been selected as the central concept of the discourse, but to find the intended antecedent, contextual knowledge about the previous discourse and world knowledge about possible semantic links are needed. Conventional surface level constructions also encode semantic-rhetorical links between propositions, paragraph structuring and topic shifting (Redeker, 1990).

Since we are interested in the exchange of information that takes place in dialogues, we make a different level of distinction, and distinguish the thematic structure of a contribution from the information structure of a contribution. The former concerns the content of the contribution in terms of what the contribution is about and what is background, the latter concerns old and new information conveyed by the contribution.

These aspects have been referred to by the terms ‘topic’ and ‘focus’, but there exists a wealth of different terminologies around these notions. They are interrelated and in simple sentences not necessarily encoded separately. This has caused confusion not only in terminology but on the conceptual level, too. Below we briefly clarify the different uses of these terms, in order to justify our contribution to the terminological diversity by our own concepts ‘Central Concept’ and ‘New Info’.

7.1.1 Topic and focus

Linguistic approach

In linguistics, the distinction has been made between *topic* and *comment* (Sgall et al., 1973), *theme* and *rheme* (Halliday, 1967), *old* and *new* (Clark & Haviland, 1977), *foregrounding* and *backgrounding*, *conscious* and *non-conscious* (Chafe, 1976). Quirk et al. (1972) talk about ‘focus’ and ‘theme’ of a sentence. ‘Focus’ is the information centre of a sentence, while ‘theme’ is defined as the first noun phrase constituent in a sentence, except if this is a certain preposed adverbial or if the constituent occurs in a thematically reorganised sentence (thematic fronting or topicalization).² This definition of a theme slightly differs from the one given in Systemic Grammar literature (Halliday, 1967), where ‘theme’ can include any first element in the sentence. ‘Unmarked focus’ contains new information, and is usually found at the end of the sentence which is the place for new information. Since ‘unmarked theme’ usually contains old information and appears in the beginning of the sentence, it is maximally distinguished from ‘unmarked focus’.

A sentence can also be thematically reorganised for discourse purposes. For instance, topicalization (*Cakes, I’ll need on Wednesday*) and clefts (*It’s the cakes that I’ll need on Wednesday*) can change the sentence structure so that the sentence gets a ‘marked theme’. In these cases, the original theme (‘I’ in the above examples) is still retained, and the sentence has an old and a new theme simultaneously. The sentence can also contain a contrasted element which is the focus of the sentence. It can appear anywhere in the sentence, being marked by intonation or focus-markers (‘only’, ‘even’). Contrasted focus contains old information, and confusion can now occur because a marked theme coincides with a contrastive focus. An unmarked theme may coincide with a contrastive focus (*Even I need cakes on Wednesday*) or it may not (*I need only cakes on Wednesday*).

²Verb initial sentences (YN-questions, imperatives) do not have a theme either.

Linguistic approaches concentrate on the surface structure of the contributions, and the definitions of ‘topic’ and ‘focus’ tend to underestimate contextual information needed in the analysis as well as the constructive nature of discourse: sentences are not produced in isolation but as a reaction to what has been previously talked about. A more discourse-oriented way to distinguish ‘topic’ and ‘focus’ can be found in Carlson (1988) and van Kuppevelt (1991) who suggest that ‘aboutness’ is defined with the help of questions that the contribution or discourse has addressed. This kind of aboutness is the ‘topic’ of a contribution or a discourse, and topical questions are what the discourse is *about*. ‘Focus’ can be defined with respect to ‘topic’ as the new material that answers the question: if a sentence XY is a reaction to a sentence XAY , a string B is new or focus, if it replaces A , otherwise it is old (Carlson, 1988). A thorough discussion on the old-new distinction can be found in Carlson (1983). This approach has been successfully applied by Vilkuna (1989) to Finnish sentence structure, and it can be combined with ‘dialogue games’ (Carlson, 1983) to cope with whole dialogues.

Topical questions are analogous to coherence relations in that they link propositions in consecutive sentences together. Of course, a mechanism is needed to determine whether and how topical questions are related to each other, i.e. to guide how coherent discourse can be built from appropriate topical questions in a sensible way. Thus they also suffer from the same drawback as coherence relations in general: unless grounded on general principles of communication, they appear random explanations of individual sentences.

NLP Approach

Previous work in natural language generation has shown the importance of ‘topic’ and ‘focus’ in different special tasks: restricting the content of the text (McKeown, 1985; Hovy & McCoy, 1989; McCoy & Cheng, 1990), determining appropriate referring expressions (Grosz et al., 1983; Reichman, 1985; Grosz & Sidner, 1986), selecting surface level expressions (Derr & McKeown, 1984), and accounting for thematically coherent discourse (Sibun, 1991). Moreover, Grosz and Sidner (1986) have pointed out that the attentional level (focussing) must be distinguished from the intentional level

(goals and intentions) in order to capture adequate explanation of discourse phenomena such as referring expressions and interruptions.

Due to the different approach and research goals in general, the concepts ‘topic’ and ‘focus’ have been used somewhat differently in the NLP literature than in linguistics. Most notably, the subtle two-way distinction has not always been made. Terminology has concentrated on ‘focus’ which can be pronominalised and which is associated with the *centre of attention* (Sidner, 1979; Derr & McKeown, 1984; McCoy & Cheng, 1990) or with *salient entities* (Grosz & Sidner, 1986), i.e. with aboutness of dialogues, rather than with contrast or newness (with intonational characteristics). An exception is Reichman (1985) who uses four-level focus assignment, depending on the influential status of the element in the currently relevant discourse context: high (pronominal reference), medium (reference by name), low (reference by description), and zero (implicit reference).

Grosz et al. (1983) distinguish between ‘centering’ and ‘global focus’, the former referring to a local focussing process whereby a single entity is identified as the most central one in an individual sentence, and the latter dealing with entities that are important to the overall discourse. They associate each sentence with a single backward-looking center which encodes the notion of centering, and a set of forward-looking centers which provide the elements to which the succeeding discourse may be linked. They show how the realisation of a backward-looking center cannot be determined solely from the syntax of the contribution, and claim that the relation between the center and its realisation is not solely semantic or pragmatic either. Rather, the basic constraint is on the speaker who should use a pronoun if the center of the current contribution is the same as the center of the previous one. Violations of this basic centering rule then direct the hearer to draw additional inferences, needed e.g. to recognise a shift in global focus.

Our approach is related to that in Grosz et al. (1983), since we also claim that the rules for global and local coherence are based on the speakers. Coherence is a result of the speakers’ compliance with communicative principles, and the inferable links between discourse referents show that these principles have been respected. We assume that

the backward-looking center, or ‘Central Concept’ as we call it, encodes the view-point from which new information is presented. It is selected so that the dialogue is coherent, based on the speaker’s communicative competence. We also deal with givenness and newness of information, which are important in the exchange of information. Global coherence is guaranteed by the speakers’ rational motivated activity.

7.1.2 Central Concept and NewInfo

In studying how coherent dialogue continuations arise from the agent’s evaluation of the previous contribution in the context, we have found the following distinction useful (Steedman (1991) also proposes the same distinction, but with different terminology):

- **aboutness**: topical vs. non-topical discourse referents, or ‘what the contribution is about’ vs. ‘what is in the background’,
- **newness**: influential status of discourse referents, or ‘what is new’ vs. ‘what is old information’.

These two dimensions are modelled with the help of the following two concepts:

Central Concept (CC): a distinguished discourse referent which is talked about in the contribution. CC is an instantiated world model concept, and it sets the ‘view-point’ for coherent continuations. All the concepts related³ to a particular CC form the background which is ‘active’ in the planning of a contribution, and those related concepts which are instantiated as discourse referents are assumed to be known to both participants.

View-points can change in the course of the dialogue, since the speaker may want to focus on something other than what has been talked about. Coherent shifts are defined by topic shifting rules, based on relations in the world model (see Section 7.2.3). A CC need not be explicitly realised in the contribution (ellipsis) or it may be referred to by a pronoun.

³Relatedness is defined by the smooth topic shifting rules, see Section 7.2.3.

NewInfo (NI): a concept or property which is new with respect to some CC. NewInfo is the information centre of the contribution, related to the main goal of the dialogue: to exchange information. It is determined by reasoning about the joint purpose and it encodes the result of the evaluation of the partner's contribution: if an initiative is taken, NewInfo is the information asked, and if a response is given, it is the piece of information provided. It can thus be described also as what Chafe (1976) calls 'conscious' information: "what the speaker assumes he is introducing into the addressee's consciousness by what he says". Notice that NewInfo of a contribution is chosen before the CC: this reflects the fact that contributions are formed around a 'point' which needs to be made known to the partner, and the CC is shifted accordingly to make the presentation of the NewInfo coherent in the dialogue context.

NewInfo is always different in different contributions, since it is not rational to carry on with a dialogue contribution which repeats the same NewInfo.⁴ As a corollary, NewInfo must be explicitly present in the contributions, and it cannot be pronominalised.

This distinction has not been singled out explicitly in NLP systems. The importance of Central Concept, or 'focus' has been acknowledged, but NewInfo is implicit: it is included in the task structure and the successive steps in the execution of a plan. However, if a system cannot rely on a detailed task structure, but needs to cope with wide contextual knowledge, as is the case in our system, the status of the exchanged information is important. NewInfo encodes the crucial part of the message to be communicated to the user, while Central Concept anchors the topic to the domain knowledge already in the context.

⁴A violation of this rationality causes the hearer to look for a reason for the repetition: apparently there is a failure in contact, perception or understanding. We do not regard explanations 'in other words' as repetitions, since they contain new information, namely the new phrasing due to another view-point.

7.2 Topic shifting

7.2.1 World and application knowledge

The world model deals with car hire and restaurant scenarios which the the application domain, a part of the Yellow Pages backend database contains information on. The application model is an application-specific part of the world model so that the system can reason about the application domain knowledge without needing to consult the database or even to know its structure. Figure 7.1 shows part of the world model used in the description of the topic shifting rules.

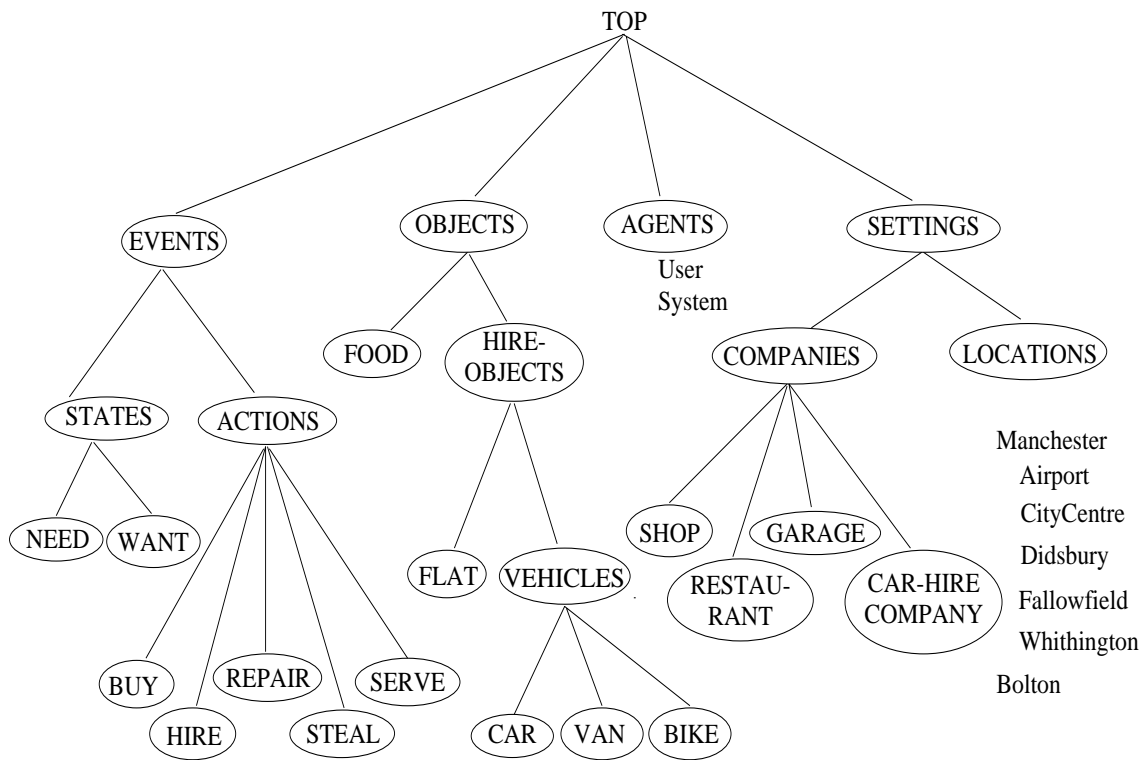


Figure 7.1: Part of the world model.

The ontology contains events, objects, agents and settings. Events are further divided into states and actions, differing conceptually in that states have two arguments corresponding to the actor and object involved in the state, while actions have three arguments: besides the actor and object they include the place where the service involved in the action can be supplied, i.e. an argument for the company that provides the service enabling the particular action.

A subclass of objects is the class of hire-objects and these include flats and vehicles. The class of vehicles contains cars, vans and bikes, differentiated with respect to different attributes like the number of wheels and the size.

Settings are divided into two subclasses: companies and locations. Companies can occur as the third argument of actions, while locations are places where events or states can take place. Companies also have properties which concern the information that Yellow Pages contain about the company (name, address, telephone, location, type, opening hours). Some properties like ‘opening hours’ are optional and need not have a value.

The model also includes the concept of a (rational) agent which can be instantiated either as a user or a system.

7.2.2 Topic shifting and rational agents

NewInfo is the information centre of a contribution. The speaker reacts to NewInfo of the partner’s contribution and either accepts it as relevant (if it addresses the evocative dimension of the previous contribution and is thematically related to the dialogue context), disapproves of it as incomprehensible or irrelevant in the context (vague, unrelated, or misunderstood) or rejects it altogether (conflict). Acceptance refers to situations where the speaker gives an appropriate answer to the partner’s question or considers the partner’s response as an appropriate answer to her own question. If NewInfo is not understood or its relation in the dialogue context is unclear, the speaker can take an initiative to clarify the contribution.

NewInfo is related to some Central Concept from which view-point the information is presented. NewInfo encodes a potential topic *shift*, since the partner can start to talk about it either by providing further information or requesting clarification on it. By taking the initiative the speaker shifts the CC to a related one (smooth shift) or to a totally new one. The system can shift CC only if the user response is vague or misunderstood, i.e. under the same conditions that it can take the initiative.

We define the following obligations that the speakers obey in order for a dialogue continuation to be coherent. These obligations are applied when the speaker determines the view-point for presenting the NewInfo, and thus they make use of contextual factors such as initiatives and the speaker's goals. However, they are of a different type from those discussed in connection with the determination of the joint purpose: while the joint purpose was based on the speaker's rationality and motivation to determine what is a good response in the context, we now deal with obligations derived from the speaker's competence to express the appropriate message adequately. 'Current CC' refers to the CC of the partner's last contribution.

1. Acceptance, shifting, maintenance and rejection of CC

- If the speaker cannot evaluate NewInfo, she shifts CC to NewInfo (clarifies the topic).
- If the speaker has the initiative and the partner has responded with a thematically related, clear NewInfo, the speaker either shifts the CC (reverts to a previous topic or introduces a new topic)⁵ or maintains the current CC (continues with the same topic), depending on whether the current CC is fully specified or not (topic is closed or not), respectively.
- If the speaker has the initiative but the partner has responded with thematically unrelated NewInfo, the speaker rejects the shift and continues with her own CC (rejects the topic).
- If the partner has the initiative and the speaker has no goals to pursue, the speaker accepts the partner's CC and continues with the current CC (accepts the topic).
- If the partner has the initiative and the speaker has her own goals to pursue, the speaker either continues with the partner's current CC (accepts the topic) or shifts the CC (reverts to a previous topic or introduces a new topic), depending on whether the partner has shifted CC to a thematically related one or not, respectively.

2. Shifting of CCs

- Topic shifting rules are based on thematic relatedness which is determined by domain-based relations.
- CC of a contribution is thematically related to NewInfo of the same contribution.
- If the current CC and the intended CC are thematically related, shift to the intended CC.

⁵There is an asymmetry between the system and the user here: the user can shift to a brand new topic, but the system is only allowed to introduce topics thematically related to previous ones.

- If the current CC and the intended CC are not thematically related, but the intended CC is thematically related to some previous CC, shift to the intended CC and make it explicit.
- If the intended CC is not thematically related to any previous CC, discard the intended CC and find another viewpoint.
- If the intended CC is not thematically related to any previous CC and there are no candidates left, introduce a brand new CC and start a new dialogue from this view-point.
- If the properties of a CC are instantiated, the CC is fully specified, i.e. the topic is closed.
- If a topic is closed, continue with a related one.

3. Thematic coherence and contributions

- Each contribution must have a CC and NewInfo.
- A contribution can have only one CC.
- A contribution can have several concepts which convey NewInfo.
- If a contribution contains several clauses, they must be thematically related to each other.

By default, the speakers continue with the same topic until it is mutually closed, i.e. understood clearly enough by both or explicitly rejected. Thus topic changes occur when NewInfo needs clarification (the evoked response is not understood or is found only partially relevant in the context), information concerning the CC has been exhausted (there is nothing else to be said about the topic), the information exchanged so far is enough to perform the planned task (the goal of the communication is fulfilled), or the speaker's interest is directed to something else.

When a topic shift to NewInfo occurs, the common terminology is to talk about 'sub-dialogues'. Since this presupposes that some kind of interruption in the main course of the dialogue has taken place, after which the dialogue must resume the original topic, we will not use the term. We do not consider topic shifts to NewInfo as deviations from the main topic, but treat them as coherent continuations of the dialogue, indeed, as signs of the negotiative nature of dialogue. The resuming of the previous topic is a consequence of the changed context in which the partner's contribution is evaluated

and a response planned.⁶ For instance, in dialogue 32, repeated here as 53, the ‘sub-dialogue’ User7-Wizard7 concerns the user’s clarification of the NewInfo given by the system (‘type of insurance service’), and once the user has cleared this up, she reports the result by providing the requested answer. The system evaluates the user’s answer, concludes that the new information is mutually known and accepted, and continues with its own goal. ‘Resuming’ of the previous topic (‘insurance service’) is a side-effect of the system’s evaluation of the user contribution in the context, based on the communicative principles that determine the appropriate joint purpose.⁷ In dialogue 33, repeated as 54, an analogous situation occurs in User2-User3, but the system is now able to give the required information without initiating any further clarifications.

- (53) User6: have you any information on car insurance.
 Wizard6: What type of insurance service do you want?
 User7: What type of services there are available?
 Wizard7: You can choose one of the following categories:
 Insurance administration services
 Insurance agents
 Insurance brokers
 Insurance companies
 Insurance consultants
 User8: Insurance agents please
 Wizard8: *Still in South Manchester?*

- (54) User1: I’m looking for a restaurant.
 System1: In which area?
 User2: What types of restaurants do you list?
 System2: Indian, Mexican, Chinese, Italian, Thai.
 User3: Try Indian in Rusholme.
 System3: *Please wait....*
 Here’s the list <>

⁶Of course, there may also be real interruptions, like for instance telephone calls or somebody rushing in and asking a question, after which the dialogue will be resumed at the original topic. We do not consider these situations, however, since human-computer dialogues tend to be continuous interactions, and interruptions like this are not recorded by the system except as a longer waiting time for a user input.

⁷The elliptical form of Wizard8 is due to the fact that the contribution is a check-question: the wizard wants to confirm that the location of insurance services is the same as that of the insurance consultants discussed earlier, see p. 150. If the exmple had occurred without prior contextual knowledge, the appropriate realisation of Wizard8 would be *In which area would you like to get the insurance agents?*

Being rational agents, the speakers do not shift topics arbitrarily. Topic shifting follows coherence requirements of the dialogue, which are rooted in the principles of ideal cooperative communication. Communicative responsiveness requires the agent to report the result of the evaluation of the partner's contribution, and consequently, the reporting is related to what the partner said, especially to the new information being conveyed. The communicative competence of the agent is shown in the way in which she presents the NewInfo of her contribution from a coherent view-point, i.e. selects the Central Concept of her message according to the dialogue context.

7.2.3 Topic shift rules

A common way to deal with topic shifting is to use a stack mechanism to encode the concept that is currently focussed on (thus the stack is called 'focus stack'). The stack may consist of simple concepts (Sidner, 1979; McKeown, 1985), or it may have a more elaborated structure in terms of 'context spaces' (Reichman, 1985) or 'discourse segments' (Grosz & Sidner, 1986). The basic idea is to store all previous foci so that the most recent is the most accessible, and also list possible focus candidates, ordered according to focussing rules (heuristics), and push and pop the stack depending on the current discourse situation. However, a stack is rather inflexible in focus management, since it requires pushing and popping of foci in a particular order in/from the stack. Dale (1988) notices that a stack of open focus spaces is inadequate when generating appropriate referring expressions, since it does not account for references made to entities which are no longer in the focus stack.⁸ He suggests that a tree-structured discourse model may be more appropriate.

McCoy and Cheng (1990) also argue that a tree structure is more flexible than a stack in managing focus shifts. They try to cover different types of focussing phenomena by introducing the notion of 'focus tree', allowing traversal of the branches in different orders. In their model, the coherence of the text is determined on the basis of the

⁸This can also be seen as a criticism of the appropriateness of structural approaches to dialogue management in general.

distance between focus nodes in the tree. As a basic rule of thumb, a shift in the ‘focus of attention’ is coherent, if the focus of each contribution in the discourse is related to the focus of the last contribution via focus shifting rules, or the information about the shift in focus is explicitly present in the dialogue in the form of a meta-comment.

The focus shifting rules refer to a model of the conceptual structure of the domain of discourse. For instance, the following discourse (their example) is based on a network of related concepts which represents the speaker’s knowledge about ‘John’.

I know John. He has brown hair and blue eyes. He likes playing football and collecting stamps.

The discourse picks out a subgraph of the conceptual model, a ‘focus tree’ which represents the focussed concepts of the discourse. This is given in Figure 7.2.

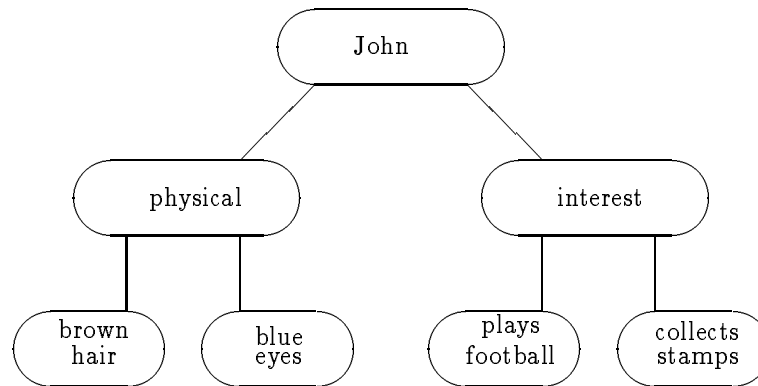


Figure 7.2: Focussed concepts in the discourse about John. Taken from McCoy and Cheng (1990).

The discourse is coherent because its parts are related to each other, and it does not mix together physical characteristics and interests. The following discourse is incoherent because it interleaves physical attributes and intrests:

I know John. He has brown hair and likes playing football. He has blue eyes and likes collecting stamps.

The focus shifting rules are expressed in terms of the types of relationships which occur in the domain. Different node types (McCoy and Cheng (1990) have ‘object’,

‘attribute’, ‘setting’, ‘action’ and ‘event’ type nodes) allow different focus shifts (see below). In generation, they provide information about whether or not a focus shift is easy to process and, consequently, whether or not the hearer will expect some kind of marker, and in analysis, they help to decide on what sort of focus shifts are likely to occur.

The rules provide only a guide as to what sort of shifts the partner will find coherent and easy to process. It is of course sometimes the case that agents will wish to shift the focus in a way which is not allowed by the focus shifting rules. A simple rule of conversation might add the special markers *oh*, *and* to the last sentence in the following example, where the focus is shifted back to the properties of John after the discourse has concerned his hobbies:

I know John. He has brown hair. He likes playing football and collecting stamps. Oh, and he has blue eyes.

We have formulated our topic shifting rules after McCoy and Cheng (1990). We have eleven different coherent shifts depending on the type of the node which is the current CC. As described in Section 7.2.1, our world model ontology contains four types of entities: objects, agents, events and settings. We do not allow dialogues to concern the agents themselves, and thus there are no shifts from this type of node. The node type ‘attribute’ does not refer to a concept, but to the property and property value of a concept (mainly information about companies). Coherent Topic Shifts are listed in Figure 7.3.

The shifts from ‘object’ and ‘attribute’ type nodes are analogous to those in McCoy and Cheng (1990), except that we have a shift to the sub- or superclass of the object as well. The shifts from our ‘event’ node subsume the shifts from their ‘action’ node. We do not have their ‘event’-type node at all since it appears vague (“shifts to action which can be grouped together into an event”). The shifts from our ‘setting’-type nodes differ from the corresponding ones in McCoy and Cheng (1990) in that they include shifts to sub- and superlocations, but not shifts to objects involved in the setting.

These domain-based topic shift rules allow us to define three types of topic shift:

Node type	Coherent Topic Shift candidates
Object:	event in which the object participates, attributes of the object, sub- or superclass of the object
Attribute:	object that the attribute describes, more specific attribute
Event:	participants in the event, location, time of the event, next action in sequence, sub- or superaction
Setting:	event that takes place in the setting, sub- or supersetting
Topic continues	

Figure 7.3: Coherent topic shifts. Modified from McCoy and Cheng (1990).

- A topic shift is *coherent* if it obeys the topic shift rules or if the topic stays the same. In general, a shift is thematically related, if the distance between the immediately previous CC and the current CC is less than two nodes.
- A topic shift is *awkward* if it does not obey the topic shift rules but the current CC is thematically related to some previous CC. In other words, there exists a path between the current CC and some previous CC, the length of which is less than a certain limit. When an awkward shift is made, the shift must be marked by a topic marker (*by the way, then, going back to, etc.*).
- A topic shift is *incoherent* if the shift cannot be justified by the topic shift rules nor related to any previous CC, i.e. there exists no path between the current CC and any previous CC. The path may be missing because of a conceptual hole in the world model (no information available), or because the topic tree is internally incoherent: the new information does not fit together with the information already instantiated in the tree. The system is not allowed to do incoherent shifts, and replanning must take place if such a shift is planned. If the user makes an incoherent shift, this usually means a new request.

7.2.4 Corpus examples

The following example clarifies what we mean by Central Concept and NewInfo. CC is bold-faced, NewInfo is in italics.

- (55)
- (1) User: I need a car.
 - (2) System: Do you want to *buy or rent* **one**?
 - (3) User: *Rent.*
 - (4) System: *Where?*
 - (5) User: In *Bolton*.
 - (6) System: Ok. Here are **the car hire companies** in Bolton:
< *list* >

The first user contribution (1) is privileged in having no thematic or information status constraints. The whole proposition is NewInfo, since all the contribution's information is new in the dialogue context. It also starts the dialogue and thus the whole proposition also introduces what the dialogue will be about.⁹ According to our definitions, NewInfo is related to CC, and thus the first contribution does not contain a proper CC or a proper NewInfo. Instead, the system needs to decide what the user wants to talk about on the basis of its knowledge about the general purpose of the dialogue: the user wants to get YP-information on some topic. In this particular case, at the beginning of the dialogue, we stipulate that NewInfo coincides with CC, and both contain the whole proposition; in this example they are instantiated to 'needEvent'.

The first system contribution (2) clarifies the vague user request. NewInfo deals with the information that the system needs to know: whether to continue with services that enable buying or with those that enable renting. CC is shifted from 'needEvent' to 'car' and NewInfo concerns the kind of event the car is involved in. The shift is coherent, since 'car' is a participant in 'needEvent', and it is motivated by the joint purpose the system has come up with (clarification of the expressed statement that the user needs a car). The user accepts this shift and provides the requested NewInfo in the

⁹There will often be a greeting exchange at the very beginning of the dialogue, but the 'body' of the dialogue starts with the description of the topic. Levinson (1983, p. 312) calls this the 'first topic slot' in his description of telephone conversations.

contribution (3). The topic continues, although now the CC is not explicitly realised but the contribution presents NewInfo only. Next the system, which has the initiative, can choose the topic. Since there is nothing more to say about the current CC ‘car’, the CC can shift. The joint purpose has resulted in a system goal where NewInfo concerns the location of hiring, and since ‘hireEvent’ is an event in which ‘car’ participates, CC can be coherently shifted to ‘hireEvent’. The CC is not explicitly expressed on the surface, since it is the NewInfo of the previous contribution. The user response (5) is analogous to (3): the user accepts the topic (‘hireEvent’) and supplies the NewInfo (‘Bolton’). Now the system can again decide the topic. This time the joint purpose has resulted in the system goal to give the list of requested car-hire companies to the user. NewInfo concerns the list of car-hire companies and the CC is shifted to ‘car hire company’. The CC participates in ‘hireEvent’ and it contains the list of service suppliers as its property.

The shifts of CC that occurred in the dialogue can be represented as the tree given in Figure 7.4. The tree is an instantiated subgraph of the world model, and its existence shows that the dialogue contributions are thematically related to each other and the dialogue is coherent.¹⁰

Dialogue (56) exemplifies topic continuation. The user keeps asking questions that concern the recently given list of restaurants, and the wizard answers accordingly. The question-answer pairs are linked together via an exchange of NewInfo about the same Central Concept. A topic shift occurs in the final pair to an unrelated concept (‘bus timetables’). Because the user has the initiative and the wizard has no goals of her own, the shift is treated as a shift to a new request.

¹⁰The compound ‘car-hire company’ can be decomposed into a conceptual cluster ‘company where car-hire event in Bolton takes place’, but our world model contains a separate concept for the compound. It was argued in PLUS that the world model should not contain this kind of compound concepts but they should be decomposed.

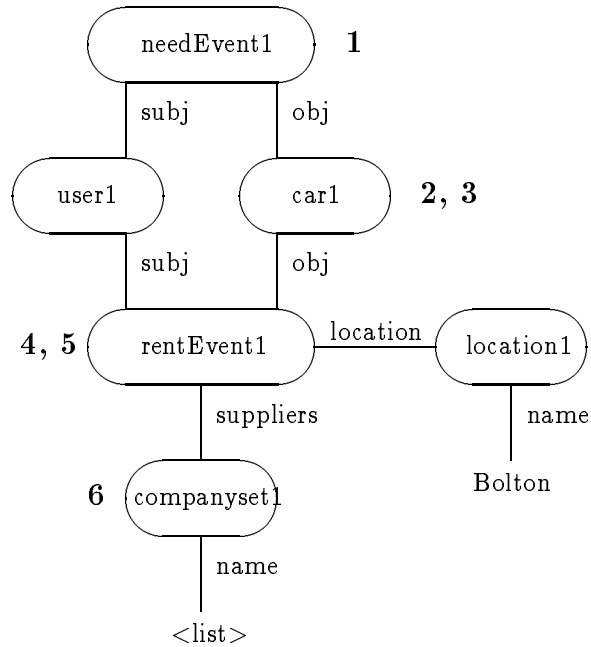


Figure 7.4: The topic tree for sample dialogue (55). Numbers refer to the contributions.

- (56) User1: are there any Mexican restaurants or Chinese restaurants?
 Wizard1: Please wait ...
 Chinese: Hong Kong Restaurant, 101 Manchester Road, Chorlton
 Mai Wah Restaurant, 485 Barlow Moor Road, Chorlton
 Mexican: La Casita, Wilmslow Road, Withington, 061-445 7643
 User2: do **they** serve vegetarian food?
 Wizard2: You should contact **the restaurants** themselves about the menu.
 User3: are **they** open after midnight
 Wizard3: **The restaurants** above do not specify their opening times;
 However, in general such **restaurants** are open until 2 am.
 User4: Do you know bus timetables?
 Wizard4: Sorry no information available.

An interesting detail is that the user refers to the CC (Mexican and Chinese restaurants) with the pronoun ‘they’, while the wizard uses a definite noun phrase. The wizard could have used the pronoun, too (“You should contact them about the menu”, “They do not specify their opening times”), but has preferred the full noun phrase to make the CC explicit and the answers more formal. Perhaps a more subtle reason also plays a role here: the wizard has based her responses on the general world model knowledge about restaurants and not on the properties of the particular restaurants listed.

Wizard2 gives compensation for the missing database information when the information is, nevertheless, understood as related to the CC, and Wizard3 gives compensation when the requested information can be given in the form of general information, cf. dialogue (13), p. 84.

In the following dialogue, the system makes contributions with compensation and the user asks yes-no-questions.

- (57)
- (1) User: can I have a list of **car hire companies** in the Entwistle area.
 - (2) System: *Where is Entwistle?*
 - (3) User: **Entwistle** is outside *Bolton*.
 - (4) System: Altogether there are *15 car hire firms* in Bolton.
Do you want a *list*, or can you be more specific in your enquiry?
[**car hire firms**]
 - (5) User: Are there **any** in Entwistle itself? [**car hire companies**]
 - (6) System: No, but you could be more specific in your requirements;
for example, the YP can supply lists of those firms which offer unlimited mileage, delivery/collection, one way rentals, cheap local rates.
[**car hire firms**]
 - (7) User: *One way rentals*, please. [**car hire firms**]
 - (8) System: *Where would you like to return the car?* [**rental type**]
 - (9) User: I want to go to *Manchester Airport*. [**rental type**]
 - (10) System: **The following companies** have branches both in Bolton and at Manchester Airport:
< list >

The topic tree for the dialogue is given in Figure 7.5.

7.3 Determining Central Concept

The main function of determining the CC is that the message must fit the dialogue context. Once the communicative goal has been formulated (so that it conforms to the obligations of sincerity, motivation and consideration), the point of view from which this information is to be presented must be decided. This means that the CC which provides the point of view must be related to the overall dialogue context. In this section we discuss how the CC is chosen by the system when it plans a response. The

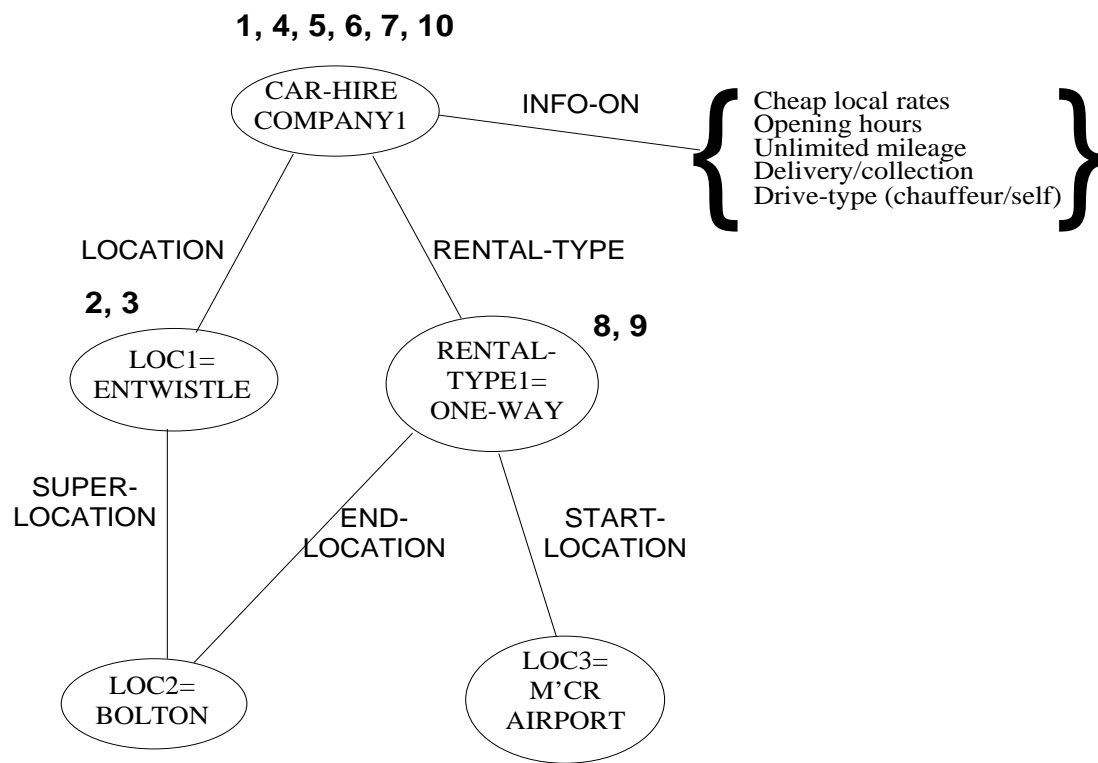


Figure 7.5: The topic tree for sample dialogue (57). Numbers refer to the contributions.

process of determining the joint purpose has resulted in a system goal and thus the system knows what is the communicative goal of the response and what is the NewInfo that is to be conveyed to the user.

The communicative goal is represented as a list of concepts and the next CC is one of these concepts. Some of the goal concepts are distinguished as NewInfo, but the CC cannot be any of these, since NewInfo is the information that is going to be introduced by the contribution, not talked about.¹¹ On the other hand, CC must be in the set of discourse referents already introduced in the dialogue, since these form the primed context in which the thematic coherence is evaluated. CC thus belongs to the intersection of the set of goal concepts and the set of discourse referents.

If the intersection contains more than one element, the alternatives are ranked according to our accessibility classification. As mentioned in Section 6.1.3, discourse referents are classified into three types: previous CCs (topics), salient and background. Their

¹¹CC cannot be either of the agents either, since the roles of the participants do not allow the user or the system to speak about themselves.

accessibility as the next CC is ranked according to the following heuristics:

1. The previous CCs have preference over salient discourse referents related to them.
2. The more recent CCs and salient discourse referents have preference over the earlier ones.
3. Background discourse referents are the least probable next CCs, although not impossible.

If the intersection of goal concepts and discourse referents is nil and there are no accessible concepts available either, a totally new goal has been formulated and a new dialogue started. Since our system is an information provider (as opposed to an expert advisor), such situations do not occur: the system can shift CC only when it can take the initiative, and this is restricted to situations where the system clarifies vagueness or offers compensation. Both situations are thematically related to the previous dialogue context.

The first sample dialogue is repeated below; CC of each contribution is marked with boldface or enclosed in brackets:

(58)

- (1) User: **I need a car.**
- (2) System: Do you want to buy or rent **one**?
- (3) User: Rent. ['car']
- (4) System: Where? ['hireEvent']
- (5) User: In Bolton. ['hireEvent']
- (6) System: Ok. Here are **the car hire companies in Bolton**:
 < list >

The relevant concepts for the planning of the first system contribution are:

```
Goal:    know(s, [wantEvent(w,u,d), disj(d,b,r),
                buyEvent(b,u,c,_), hireEvent(r,u,c,_), car(c), user(u)])
Goal Concepts: wantEvent(w,u,d), buyEvent(b,u,c,_),
                hireEvent(r,u,c,_), car(c), user(u)
Discourse referents: needEvent(n,u,c), car(c), user(u)
NewInfo:  disj(d,b,r)
Previous CC: needEvent(n,u,c)
```

The new CC must be one of the discourse referents, and it must belong to Goal Concepts. In this particular case, their intersection is $\{\text{user}(u), \text{car}(c)\}$. Both of these candidates are participants in ‘needEvent’, and thus offer a coherent shift from the previous CC. However, the roles of the user and the system as an information seeker and an information provider do not allow the user to talk about herself, nor the system to assume that the user will want to talk about herself, and thus $\text{user}(u)$ is discarded as a possible next CC. The only possibility left is $\text{car}(c)$. CCs can be pronominalised, and so the CC appears as a pronoun in the system response. Because the CC occurs in the scope of an intensional predicate¹², it is pronominalised as *one*.¹³

The user provides the system with the requested information, and the system continues to fulfil the original request. The second system contribution has the following context:

```

Goal:    know(s, [location(r, _), hireEvent(r, u, c, _),
               car(c), user(u)])
Goal Concepts: location(r, _), hireEvent(r, u, c, _), car(c), user(u)
Discourse referents: needEvent(n, u, c), car(c), user(u),
                    wantEvent(w, u, r), hireEvent(r, u, c, _)
NewInfo: location(r, _)
Previous CC: car(c)

```

The intersection of discourse referents and goal concepts gives the following set of possible next CCs: $\{\text{hireEvent}(r, u, c, _), \text{user}(u), \text{car}(c)\}$. The concept of $\text{user}(u)$ is discarded as above. So is $\text{car}(c)$, since there is nothing more to be said about the car, and also since CC must be related to NewInfo which in this case concerns the location of hiring and not that of the car. The shift from $\text{car}(c)$ to $\text{hireEvent}(r, u, c, _)$ is coherent and ‘hireEvent’ is selected as the next CC.

The question is formulated as an ellipsis. Elliptical contributions are the default since the dialogues are exchanges of new information, and the speakers react to and present NewInfo only, if CC is contextually recoverable. In this case, CC need not be made explicit since the CC is clear and the contribution conveys the message without false implicatures (i.e. without referring to the location of other discourse referents). It is

¹²Intensional predicates are predicates whose object need not be referential, such as ‘want’ or ‘need’.

¹³In other contexts CC is pronominalised as ‘it’ or ‘they’, given that other constraints governing pronominalisation are fulfilled.

not probable that the hearer will analyse the question as referring to the location of ‘wantEvent’, since this is only background information. It is more probable that the question will be understood as referring to ‘needEvent’ which is one of the previous topics, but because this is a closed topic (fully specified on which there is nothing more to be said), it is discarded as well. Finally, the location cannot concern the location of the car, since the semantics of ‘location’ requires that it is applied to extensional objects only, but ‘car’ has been introduced as an intensional object.¹⁴

The user again supplies the requested information, the system evaluates its unfulfilled goals in the changed dialogue context, which results in the goal of listing car-hire companies. The context for the last system contribution is as follows (only the main goal attitude is given):

```

Goal:    know(u,ypinfo(chco,<list>),
         carHireCompany(chco]))
Goal Concepts: ypinfo(chco,<list>), carHireCompany(chco)
Discourse referents: needEvent(n,u,c), car(c), user(u),
                    wantEvent(w,u,r), hireEvent(r,u,c,_), location(r,bolton)
NewInfo: ypinfo(chco,<list>)
Previous CC: hireEvent(r,u,c,_)

```

The intersection of discourse referents and goal concepts is now nil. However, the previous topic `hireEvent(h,u,c,_)` has activated the concept of `carHireCompany` which, although not explicitly introduced, is an accessible concept since it is thematically related to a previous topic (the world model shows that it is one of the participants in the hire-event). Being the only alternative, this is selected as the next CC. The resulting tree of CCs was presented in Figure 7.4.

7.4 Conclusion

A coherent dialogue is easy to follow and clear in its goals. Since our system is not task-oriented in the sense of having well-specified tasks to perform, it relies on communicative knowledge and coherence defined with respect to the domain.

¹⁴It occurs as an object of an intensional predicate ‘need’. However, we do not go into details of the distinction between intensional and extensional predicates or discourse objects here.

We distinguish between Central Concept and NewInfo on the basis of their information status in the dialogue context. The Central Concept is the ‘topic’ that the dialogue is about, while NewInfo encodes the focussed information about the topic which is exchanged in the dialogue. Central Concept can be implicit in the surface contribution or referred to by a pronoun. NewInfo must always be explicit in the surface contribution and it cannot be pronominalised.

The CC takes care of local coherence in the dialogue: it provides the coherent view-point for presenting NewInfo. The principles that govern the acceptance, maintenance, shifting and rejection of CCs are derived from the speakers’ rational agenthood. Coherent shifts of CC in a dialogue are based on domain relations which provide expectations about the likely next CCs both in analysis and generation. If there are multiple candidates for a next CC, a heuristic ranking of discourse referents is used to select one.

NewInfo is important, because the planning of the realisation of a system response starts from it: NewInfo is the main information to be communicated to the user. Central Concept provides the view-point from which the NewInfo is presented to the partner, and it is important in relating the contribution to the previous discourse. Especially, CC is used to:

1. constrain the amount of explicit information that is included in a contribution to communicate the message,
2. distinguish coherent continuations from awkward and incoherent ones; the latter are then marked with an explicit marker (*by the way, and then, to go back to, etc.*),
3. determine reference,
4. generate referring expressions,
5. analyse and generate elliptical contributions.

Chapter 8

Explicitness and Ellipsis

When planning system responses, two independent tasks seem to be important: to determine appropriate information that is to be communicated, and to choose the concepts that must be explicitly expressed in the surface contribution to communicate the goal. As argued above, our system attempts to overcome the boundary problems with planning and realisation by using communicative obligations throughout the generation. The system goal is formulated as a result of the evaluation of the user contribution, but the formulated goal is further specified with respect to the system's application knowledge, communicative obligations and surface realisation criteria. The output of the planner is a fully determined semantic representation of the system response. Explicitness is part of the specification of the goal, and related to the communicative principles that govern the specification in general. In this chapter we discuss the notions of explicitness and implicitness in the planning of system responses, and how this distinction is used in the determination of elliptic contributions.

In Section 8.1, we review some previous work on explicitness, especially from the point of view of generation. In Section 8.2 we discuss explicitness and ellipsis in the light of communicative competence of the speakers and define two principles of communicative competence: the 'Responsiveness Principle' and the 'Minimalism Principle'. We also specify four criteria for determining the relevance of a concept in the dialogue context. In Section 8.3 we distinguish between explicitness and elliptical contribution and discuss

the notion of ‘realisable concept’. Finally, in Section 8.4 we present an example of how these concepts and principles are used in the planning of a surface level expression for a system goal.

8.1 Explicitness in generation

A common heuristic in earlier generation systems is to connect explicitness to the user’s knowledge. The facts that the user already knows can be left out (Mann & Moore, 1981; Weiner, 1980; Appelt, 1985), or alternatively, concepts that are known to the hearer should be used in explaining unfamiliar facts (Wallis & Shortliffe, 1984). The importance of focus and topic in determining relevant information has also been acknowledged (Grosz & Sidner, 1986; Hovy & McCoy, 1989), as well as rhetorical relations that hold between text units (Hovy, 1988a; McKeown, 1985).

Conceptual specifications of text units mostly correspond to propositions that can be realised as clause-like chunks. The problem of explicitness and implicitness is thus reduced to the building of a coherent plan from the individual propositions: sorting out which propositions need to be included in the plan and which ones can be left out. This view is much simplified: whether a fact is explicitly expressed on the surface level or not, is not only a matter of deciding whether the hearer knows the fact or not, but also a result of a complex reasoning process with respect to the context. As pointed out by Horacek (1992), the underlying content of a text cannot be expressed as a set of composable facts, since the facts have relations and dependencies which affect the surface presentation. His examples include grouping phenomena like quantification implied by assertions referring to individual objects in the domain¹, lexicalization where the lexical repertoire sets constraints on the presentation of facts, and conversational implicature which arises e.g. in the context of causal relations which raise the choice of mentioning the facts explicitly or leaving them implicit for the hearer to infer. Horacek (1992) argues in favour of a more flexible approach where one makes the relations

¹For instance, if *X1* is a single room and *X2* and *X3* are not, then in the discourse universe that consists of the rooms *X1*, *X2* and *X3*, it is appropriate to say that *X1 is the only single room*.

between text structure elements and the arguments underlying the text portion explicit. The system's communicative intention is represented in the 'argumentative structure' which is "concise, redundancy-free, and entirely explicit", and it is modified in the course of generation to yield 'text structure', from which surface text is finally produced. This approach is similar to ours, although we talk about the system's communicative goal and the semantic representation of the goal.

Reiter (1990) also addresses the problem of explicitness/implicitness by showing how the choice of a referring expression influences the implicatures that the user is able to draw. For instance, if the speaker intends to warn the hearer about a dangerous water animal, she can rely on the hearer's knowledge that sharks are dangerous animals and select the expression 'shark' instead of 'dangerous fish'. If, on the contrary, she prefers 'dangerous fish' to 'shark', the implication is that the fish is dangerous, its type is unknown, but at least it is not a shark.

Both Horacek and Reiter emphasise the importance of conversational implicatures in deciding which surface level representations are most appropriate for conveying the intended meaning. Another view-point is presented by Rubinoff (1992), who approaches the explicitness problem by considering the interaction between content planning and linguistic realisation. He describes a system in which the linguistic component provides feedback to the planning component in terms of annotations that describe the effects and consequences of particular linguistic choices. However, the disadvantage of the approach is its reliance on the linguistic component to identify the relevant effects: lack of a conceptual model and any inference system makes it difficult to deal with information that is not directly associated with particular linguistic units, but rises from the beliefs and intentions of the speaker.

Explicitness and implicitness also depend on the speakers' mutual roles and their expertise level with respect to the domain. As exemplified by Hovy (1988a), different combinations of pragmatic goals like the speakers' opinions of the topic and their relative social status generate different texts which vary in their explicitness; a combination of conflicting goals (antagonist superior with a low esteem of the topic of the conversation) actually suppressed all the text. On the other hand, Paris (1988) found that

domain experts and novices exploit different strategies when explaining how something works: experts concentrate on the functionality of the system while novices deal with the structure. She concluded that this is due to the fact that experts can infer the structure from the function, and thus descriptions of the structure need not be explicit. In our dialogues, the roles of the user and system are fixed as information seeker and provider, respectively, and their expertise level is assumed constant as well (the system is an expert on the application model and how to access information in it, while the user is an expert on her task which the system is not aware of). Thus we do not discuss explicitness related to user modelling.

Finally, explicitness and implicitness are also connected to the actual lexical items used in the contribution. For instance, the choice between ‘watch’ and ‘look at’ depends on whether the visual stimulus is changing or static, while the different lexemes for ‘smile’ (‘grin’, ‘simper’, ‘smirk’) carry various associative meanings (Stede, 1993). Since we are not dealing with lexical choice and stylistic aspects of language, we acknowledge these problems, but do not pursue them further.

8.2 Explicitness and cooperative communication

8.2.1 Responsiveness and Minimalism Principles

Consider the following utterance:

(59) 6.

Obviously, a quantity of something has been communicated, though the utterance itself contains nothing that would give a clue to what the cardinal number refers to: the exact reference of the something is implicit. However, the hearer is able to reconstruct the relevance of the response on the basis of the context in which it occurs: what is being talked about (e.g. car hire companies), what is the NewInfo of the previous contribution (e.g. cardinality of a set of car hire companies), and what intentions and expectations

have been raised in the dialogue (e.g. intention to know how many car hire companies there are in a particular place and expectation to be supplied with the number).

The same factual meaning could have been conveyed had the speaker phrased the utterance in another way. However, a different set of intentions and expectations would have been addressed. For instance, in example (60) the response also includes the positive answer *Yes*, since the simple cardinality would not address the explicit surface question about the existence of car hire companies,² and in example (61) the response (Wizard14) is appropriate because the speaker has asked similar questions about car hire companies in different places in succession.

- (60) U3: Are there any car hire companies at Manchester Airport?
S3: Yes. 6.

- (61) Wizard12: Would you like the list of car hire companies in Salford?
User13: No thanks. Are there any car hire companies Rochdale?
Wizard13: There are 5 listed. Do you want see them?
User14: No. How many firms are there at Manchester airport?
Wizard14: There are 6 car hire firms.

Example (62) below, analogous to example (52) in Chapter 6, exemplifies the relation between explicitness and implicitness from the point of view of conveying correct implicatures.

- (62) User3: Can you give me the address of a Mexican restaurant in Withington?
S3a: La Casita, 406 Wilmslow Road, 061-434 2498.
S3b: There is only one Mexican restaurant in Withington:
La Casita, 406 Wilmslow Road, 061-434 2498.

Ignoring the fact that (User3) can also be understood as a real factual question about the system's capability (and the response "yes" would thus be appropriate), we assume that it is interpreted as a conventionalised request which initiates a database search, and that the result of this search is the information about 'La Casita'. The response (S3a) is thus accurate, and, indeed, in accordance with the user request which concerns

²The system does not list the details of the companies because there are so many of them; instead it could initiate a question (*Do you want to see them?*) which would enable the user to request to see the list or continue with something else.

a restaurant. Notice that the response is elliptical and gives only the new information; the fact that the restaurant is Mexican and located in Withington is provided by the immediate context. However, if the system has reason to believe that the user wanted to get a random name from a *set* of Mexican restaurants in Withington (e.g. the previous context has dealt with Mexican restaurants in the Manchester area and the user has picked up one particular sublocation in Manchester), but the system only knows of *one* Mexican restaurant in Withington, the response is not free from false implicatures. It supports the implication that La Casita is one of the several Mexican restaurants in Withington, and thus violates the principle of Sincerity, see Section 5.2.4: the system should not give information that is not true with respect to its knowledge base. The response (S3b) makes the system knowledge explicit by the lexical focussing element “only”, and thus prevents the user from entertaining false conceptions about the system’s knowledge (i.e. about the YP information). The response (S3b) is further supported by the obligation of Consideration, which requires the system to provide information which is useful in that it enables the user to behave in a rational way and prevents her from getting false information.

Explicitness and implicitness are thus not only tied to the speaker’s goal to communicate NewInfo, but also to her knowledge about the dialogue context, especially to the knowledge about what has been talked about and which goals have been evoked by the previous contributions, and to her communicative competence to plan a surface response which conveys relevant information. On the one hand, the result of the evaluation of the partner’s contribution is to be reported, and on the other hand, the reporting must be appropriate in the context: the response may only need to convey the new information (example (59)), but it must also address the expressive and evocative intentions of the previous context (examples (60) and (61)), and must prevent the user from drawing false implicatures (example (62)).

The speakers behave as rational motivated agents, and trust the partner to behave similarly. We summarise the communicative competence of the speaker with respect to the explicit/implicit distinction in the ‘Responsiveness Principle’ and the ‘Minimalism Principle’.

The Responsiveness Principle:

Report the new information that results from the evaluation of the partner's contribution.

The Minimalism Principle:

To communicate new information, add contextual information only as much as needed in order to

- convey the whole goal,
- to avoid false implicatures,
- and to obey syntactic constraints.

The Minimalism Principle resembles the Gricean maxim of quantity: make your contribution as informative as is required for the current purposes of the dialogue, but not more informative than is required. However, it emphasises the information content of a contribution from the view-point of the overall purpose of dialogues: to exchange information. The minimum to be communicated to the partner is NewInfo, the information centre of the contribution, and the context is included in the contribution only if NewInfo cannot be successfully communicated as such.

8.2.2 Relevance criteria

The process of determining the joint purpose results in the system's own goal. The content of a communicative goal is expressed as a list of concepts which must be explicitly or implicitly communicated to the hearer. Each concept that is explicitly expressed on the surface level must be justified as being relevant for the communication of the intended meaning; if the concept fails to be relevant, it is left out. On the other hand, if the successful communication of the goal requires that a concept not appearing in the goal must also be communicated to avoid false implicatures, this concept is included in the message as well. We assume that in information-seeking dialogues the speakers want to exchange information efficiently, and consequently, only

the necessary concepts will be realised. However, if the system responses should be verbose, for stylistic or other reasons, the planning process could incorporate other obligations that would take care of such extensions.³

Relevance is defined with regard to the inferences that the hearer can draw from the contribution. The hearer is free to draw implicit conclusions, but the speaker must ensure that these do not contradict with what she intends to convey, i.e. that the hearer does not draw false implicatures. As pointed out by Joshi et al. (1984), false conclusions are blocked only to the extent that they are relevant in the communication: there is no need to prevent all possible conclusions which are not currently focussed on or in active memory.

The explicit concepts are selected with respect to the intentions and beliefs that would be expressed and evoked if the concepts formed the semantic representation of the goal. Each explicitly expressed concept is a partial realisation of the communicative goal, and thus each explicit concept must be relevant in the communication. We define a concept to be ‘relevant’ in the dialogue context if it fulfills four relevance criteria (see below) which serve as constraints on the appropriateness of the realisation. Contrary to coherence, which was based on the relations that hold in the domain knowledge, we consider relevance as directly based on the speakers’ communicative knowledge.

Reiter (1990) gives three constraints for generating successful descriptions based on a user’s domain knowledge:

- *accuracy*: the utterance should be truthful,
- *validity*: the utterance should trigger the desired inferences in the hearer,
- *freedom from false implicatures*: the utterance should not lead the hearer to draw incorrect conversational implicatures.

A particular lexical item is selected in the description, if the item satisfies the three constraints so that the facts to be communicated to the hearer are appropriately included in the expression. A ‘fact’ refers to an attribute associated with the object, and

³We can also think of the planning process being switched to a verbose mode which realises all the concepts mentioned in the goal (cf. Moore & Paris, 1993).

its successful communication depends on the user's domain knowledge. For instance, if the speaker wants to refer to a dangerous water animal, the description 'shark' is sufficient, if the hearer knows that sharks are dangerous water animals. If the speaker had used 'dangerous fish' instead, the same hearer could conclude that the animal is not a shark: if it were, the system would have used the most accurate description 'shark'.

If we substitute 'contribution' for 'utterance' and 'attitude' for 'fact', we can loosely follow Reiter (1990) and define the following criteria for determining the relevance of a contribution and the concepts included in it.

Relevance Criteria:

1. **Accuracy:** the contribution must accurately represent the speaker's goal, i.e. the concepts must express the intended content truthfully.
2. **Consistency:** the contribution must be internally consistent, i.e. the set of expressive and evocative attitudes carried by the contribution must not be contradictory.
3. **Validity:** the contribution must be valid in the dialogue context, i.e. indicate that the partner's evocative attitudes have been appropriately addressed.
4. **Freedom From False Implicatures (FFI):** the contribution must not trigger unwanted implicatures, i.e. the content must not evoke attitudes which the speaker is not able to support.

The criterion of Consistency is added to prevent the system from planning a contribution which would be internally contradictory. Its usefulness is connected to the multifunctionality of the contributions. For instance, contribution (63) with the communicative intentions to give an answer and compensate the negative result by providing helpful general information is fine. Contribution (64), however, conveys contradictory information: the speaker expresses that she both knows the opening hours and does not know them.⁴ The criterion of Consistency rules out this kind of contribution, since the attitudes are contradictory.

⁴A less certain contribution would of course be acceptable: *Usually the restaurants are open till 2am, but please contact the restaurant about the opening hours.*

- (63) No information on the the opening hours, but usually such restaurants are open till 2pm.
- (64) The restaurant is open till 2am, but please contact the restaurant about the opening hours.

The criteria of Accuracy and Consistency take care of the internal coherence of the contribution while the criteria of Validity and FFI take care of the contribution's coherence in the dialogue context. The criteria of Accuracy and Validity guarantee that the correct inferences are included in the set of communicated concepts, while the criteria of Consistency and FFI guarantee that only the correct inferences are in the set. The inferences are based on the system's domain knowledge and communicative principles.

The Relevance criteria resemble the obligations of Sincerity (Accuracy), Motivation (Consistency and Validity) and Consideration (FFI), and we can, in fact, regard them as the same obligations. The reason why they are listed here as different criteria is that now the obligations are applied to the goal from the interpretation view-point as if the goal were interpreted by the user. The joint purpose, and consequently, the system's communicative goal, is based on the application of the principles when the system reasons about the appropriate response from its own view-point. To ensure that its goal is interpreted by the user in a similar although not necessarily exactly the same way, the system must know what kind of implications the goal produces in the context and how the interpretation of the goal would change the context if the goal is expressed by a particular surface contribution. The planning of a semantic representation for the goal (from which the surface string will be formed) can be understood as a process whereby the system produces its own interpretation of the goal that it intends to convey, thus updating the context and 'checking' that the intended response is indeed appropriate in the context.⁵

The same communicative principles should, of course, operate in the interpretation

⁵The checking algorithm is slow, especially if there are several concepts and attitudes to be checked. However, as mentioned in Chapter 5, the speed of the algorithms is not the main concern of the thesis, but is left for future research.

and generation of a contribution, given that the communicative knowledge is general knowledge about how to evaluate the partner's contribution in a rational and cooperative way. The fact that we use the same principles in the formulation of the system goal and the planning of its surface realisation is considered an important feature that dissolves the distinction between strategic and tactical generation and helps to bridge the 'generation gap'.

The set of relevant concepts always includes NewInfo, but the speaker must also make sure that the user is not misled by false implicatures. The contribution may thus contain redundancy with respect to the context, if this guarantees that message is appropriately conveyed. The planning of a contribution thus does not aim at finding the smallest set of concepts that would describe the communicative goal, but rather, to find *some* set of concepts which is consistent with the contextual requirements: it contains the relevant concepts to be communicated, but does not evoke any unintended implicatures.

As pointed out by Reiter (1990) as well as Dale and Haddock (1991), the problem of finding the minimal set of facts that will convey the intended meaning is computationally intractable. Also, from the human processing point of view it appears that some kind of redundancy is important in messages: people, for instance, distinguish objects by using more discriminating attributes than is necessary. Redundancy seems to create coherence and also to help to guarantee that the intended meaning is indeed transmitted even if some other part of the message is lost or misunderstood. Mooney et al. (1991) argue that in explanation generation it is understandable that already known material is avoided if the explanation is reasonably short, but in naturally occurring explanatory texts, repetition actually serves rhetorical as well as information content purposes.

8.3 Ellipsis

8.3.1 Definitions

We make a distinction between explicit and implicit information on the one hand and between elliptical and complete sentences on the other hand. Explicitness and implicitness deal with the relevant information to be communicated to the partner, while ellipsis is based on the system's grammatical knowledge: a contribution is elliptical if it does not form a syntactically complete sentence. However, a contribution is explicit with respect to a piece of relevant information if the corresponding concept is lexically realised in the contribution, while a contribution is implicit with respect to a piece of relevant information, if the corresponding concept is to be recovered on the basis of contextual and world model interpretation of the contribution. The definitions are as follows:

- **An explicit concept** is a relevant concept which is lexically realised on the surface level. NewInfo is always explicit.⁶
- **An implicit concept** is a relevant concept which is not lexically realised on the surface level, but can be unambiguously inferred from the context.
- **Ellipsis** denotes syntactic incompleteness. A sentence is elliptical if some of the syntactically obligatory arguments of the main verb are not lexically realised (*Rent.*), or if the sentence does not contain a main verb (*In Bolton. Where? 6*).

The planning of explicit vs. implicit responses is governed by communicative principles, but the possible elliptical sentences are subject to linguistic constraints which thus ultimately determine the ellipsis generation. A concept which could be implicit may in fact appear explicitly in the surface contribution, if it is required by syntactic constraints. For instance the verb 'serve' requires a syntactic object, and thus the

⁶If NewInfo is unrealisable, replanning must take place since the system is unable to express the result of the evaluation.

realisation of the concept ‘hot and spicy food’ cannot be omitted in example (65) if ‘serve’ is lexically realised on the surface level.⁷

- (65) U: Which serve hot and spicy food?
S: * Indian, Mexican and Thai restaurants serve \emptyset .

On the other hand, implicit responses need not be elliptical, since implicit concepts can also be conveyed with the help of particular lexical units (especially with ‘pragmatic particles’), syntactic constructions and established context. For instance, in example (66), both the elliptical response (S1a) and the full-sentence response (S1b) implicitly express that the number of car hire companies in Bolton is two: (S1a) lists the car hire companies,⁸ while (S1b) uses the definite noun phrase “the following car hire companies” to emphasise the exhaustiveness of the list.

- (66) User: Are there any car hire companies in Bolton?
S1a: Budget Rent-a-Car
Bradford House 287-289 Manchester Road
Bolton
0204 391611

Hertz
Manchester North branch
Bolton
061-273 8884
S1b: The following car hire companies are in Bolton:
Budget Rent-a-Car
Bradford House 287-289 Manchester Road
Bolton
0204 391611

Hertz
Manchester North branch
Bolton
061-273 8884

The responses also exemplify how syntactic constructions convey implicit information. The user request carries the expressive attitude about the *existence* of car hire com-

⁷The elliptical response *Indian, Mexican and Thai restaurants* would be appropriate, since NewInfo (the types of restaurants that serve hot and spicy food) is clear in this context.

⁸If the system had known of other companies, it would have given their names as well according to the obligation of Sincerity and Consideration, cf. argumentation in example (52).

panies in Bolton, and thus a response which would only give the number of car hire companies (*S1c: Two.*) is inappropriate, since it would not address the explicit expressive attitude of the user contribution. In response (S1b), this is addressed by explicitly conveying the existence of the companies, but in (S1a) also this piece of information is implicitly conveyed: the listing of the car hire companies is a proof of their existence.

8.3.2 Realisable concepts

A concept is ‘realisable’ on the surface level, if there exists a mapping via a conceptual lexicon from the concept to a semantic predicate (quasi-logical form representation). The mapping from the qlf-predicate to a lexeme and to a surface level word is done by the surface generator on the basis of its linguistic knowledge. Different types of mappings between world model concepts, qlf-predicates and lexical entries are given in Figure 8.1. The arguments in the concepts and qlf-predicates denote the indices of the entities and they must match in the way shown in the mappings.



CONCEPTUAL LEXICON		LINGUISTIC LEXICON		
WORLD MODEL CONCEPT		QLF-PREDICATE		LEXEME
car(C)	\longleftrightarrow	car(C)	\longleftrightarrow	car
hireEvent(H,A,O,L)	\longleftrightarrow	rent(H,A,O)	\longleftrightarrow	hire
			\longleftrightarrow	rent
buyEvent(T,X,Y,Z)	\longleftrightarrow	buy(T,X,Y)	\longleftrightarrow	buy
	\longleftrightarrow	sell(T,Z,Y)	\longleftrightarrow	sell
financialInst(B)	\longleftrightarrow	bank(B)	\longleftrightarrow	bank
riverPart(B)	\longleftrightarrow			

Figure 8.1: Mapping types between world model concepts, semantic predicates and lexemes

The ambiguous mapping between the concept `buyEvent(−,−,−,−)` and the predicates “buy” and “sell” is called ‘conceptual synonymy’: the concept represents an entire

transaction involving two agents and the thing which is exchanged.⁹ The mapping between `financialInst(_)/riverPart(_)` and “bank” is ‘lexical ambiguity’. The mapping between the qlf-predicate “rent” and the lexemes “hire/rent” exemplifies ‘lexical synonymy’. This ambiguity does not pose problems for the dialogue manager, since the reasoning component only knows the conceptual lexicon mappings, and the mapping from `hireEvent(_,_,_,_)` is unambiguous to the qlf-predicate “rent”.

The conceptual lexicon provides information not only about mappings between concepts and NL content words, but also about how to resolve conceptual synonymy. Of course, such resolution is not always necessary: the system need not be able to reason about the distinction between two realisational possibilities from the point of view of interpretation, so that the system can ‘understand’ more than it can produce. We do not go into details, but simply assume that the necessary distinctions can be made on the basis of contextual information. For instance, the choice between “buy” and “sell” is based on the view-point from which the concept `buyEvent(_,_,_,_)` is discussed in the dialogue context. As shown in Figure 8.1, conceptual lexicon maps the second argument of `buyEvent(_,_,_,_)` onto the subject of “buy” but onto the beneficiary of “sell” (which would be expressed by a prepositional phrase ‘to someone’ in the surface contribution), while the last argument is mapped onto the subject of “sell” but onto the source of “buy” (which would be expressed by a prepositional phrase ‘from someone’ in the surface contribution).¹⁰ If the second argument of `buyEvent(_,_,_,_)` is instantiated but the fourth one is not, the preferred lexical choice is “buy”, because this allows the system to generate an active sentence instead of a passive one.¹¹ If both arguments are instantiated, information about the previous surface contributions can be used.

⁹A better name for the concept onto which “buy” and “sell” are mapped would be ‘transaction’, but here we follow the PLUS World Model and Conceptual Lexicon (Cavalli et al., 1992b).

¹⁰The first and third arguments of `buyEvent(_,_,_,_)` are mapped onto the event variables, and the objects of the predicates, respectively.

¹¹It is assumed that a passive sentence is preferred if the subject-argument of the predicate is uninstantiated. Of course, other contextual factors may require the choice to be ‘sell’ (e.g. the topic of the dialogue concerns selling rather than buying), when more sophisticated linguistic rules would be needed to enforce a passive sentence instead of an active one.

The conceptual lexicon also contains information about syntactic well-formedness constraints of different lexemes, such as that the verb ‘buy’ subcategorises for a subject and an object and that a full sentence requires them both to be present. We refer to the PLUS Conceptual lexicon for further details (Cavalli et al., 1992b).

8.4 Examples

In the analysis of a user contribution, every realised qlf-predicate is an explicit concept which appears in the content of some explicit expressive attitude. A contribution may also give rise to concepts which are revealed in the interpretation process itself, as a result of fitting the contribution into a particular context. These concepts are not tied to any syntactic-semantic expressions of the surface contribution, but to the whole contribution via the implicit (expressive and evocative) attitudes that the contribution has given rise to.

For instance, user contribution 67 has the semantic representation given below.

(67) I need a car.

```
[i(_u),num(_u,sg),case(_u,nom),def(_u),need(_n,_u,_c),
  stype(_n,finf),tense(_n,pres-time),num(_n,sg),pers(_n,st),
  forsome(_c),car(_c),num(_c,sg),case(_c,acc),kind(_c,countn)]
```

The qlf-predicates are mapped onto world model concepts via the conceptual lexicon, and the concepts are instantiated as discourse referents in the Context Model, if not already present there. The explicit expressive attitude concerns the user’s statement that she needs a car, and the explicit concepts are: *user(u)* (the first person singular pronoun is interpreted as referring to the user), *needEvent(n,u,c)* and *car(c)*.¹²

¹²The indices refer to the appropriate entities in the Context Model. We have made a simplification here in that the intensional object ‘car’ is expressed as an extensional discourse referent in the Context Model, which is of course semantically inappropriate.

Explicit attitudes are assumed to be known by both partners (because the participants speak the same language and they are cooperative agents, the interpretation of explicitly expressed concepts is assumed to result in the same set of attitudes).

When the interpretation continues, the reasoning brings in new concepts. In this particular example, the system assumes that the user wants to have a car, and the content of the implicit expressive attitude thus includes the concepts `wantEvent(w,u,h)` and `haveEvent(h,u,c)` (cf. the expressive attitudes given for the contribution in Figure 8.2 below). However, these concepts are not added to the set of discourse referents. This is because implicit attitudes represent the system's private interpretation of the user contribution and this may be incompatible with what the user intended to express: e.g. the user may be looking for information on how to get to the airport, not how to get a car.

In the generation of a system response, the process of determining the joint purpose selects the relevant concepts to be conveyed to the user. These concepts are considered implicit until they are realised lexically on the surface level. Every concept that is realisable can be explicit, and those concepts which cannot be realised can only be implicit. If a relevant concept cannot be lexically realised (i.e. it cannot be explicit), but it cannot be inferred from the context either (i.e. it cannot be implicit), the goal contains information that obviously cannot be conveyed to the user. Replanning can take place: e.g. a possible compound word must be expressed by its parts ('car hire company' > 'a company which hires out cars') or the whole goal must be revised (instead of asking if the user wants a chauffeur-driven or a self-drive car, the system can ask if the user can drive). The information may also become inferable if the context is augmented with appropriate concepts (for instance, 'one-way rental' can be explained by introducing the concepts of travelling, source location and destination location). Constructive dialogue management actually supports the augmentation of context so that a richer set of implications can be derived on the basis of introduced discourse referents and other contextual information. Of course, the augmentation requires communicative competence from the speaker, since the richer context usually also allows a larger set of unwanted implicatures to be drawn.

All the concepts that occur in the content of the evoked system intentions are called ‘Goal Concepts’, since they express the content of the system goal. One of the Goal Concepts is CC, the discourse referent which is talked about, and NewInfo is also distinguished. Examples (68) and (69) exemplify these concepts in two goals.

- (68) ‘The system wants to know whether the user wants to buy or rent a car’
`want(s, know(s, [wantEvent(w, u, d), car(c), user(u),
hireEvent(r, u, c, _), buyEvent(b, u, c, _), disj(d, b, r)])))))`
Goal Concepts: `wantEvent(w, u, d), car(c), user(u), disj(d, b, r),
buyEvent(b, u, c, _), hireEvent(r, u, c, _)`
NewInfo: `disj(d, buyEvent(b, u, c, _), hireEvent(h, u, c, _))`
Current CC: `car(c)`
- (69) ‘The system wants the user to know that there are 12 car hire companies
‘in Bolton’
`want(s, know(u, [cardinality(setOf(co, _cos), 12),
carHireCompany(co), location(co, bolton)]))`
Goal Concepts: `cardinality(setOf(co, _cos), 2), 1
carHireCompany(co), location(co, bolton)`
NewInfo: `cardinality(setOf(co, _cos), 2) 1`
Current CC: `carHireCompany(co)`

Example (68) can be studied in the following contexts:

- At the beginning of the dialogue, the user contribution is *I need a car* with the expressive and evocative attitudes as described in Figure 6.7, repeated here as Figure 8.2.
- At the beginning of the dialogue, the user contribution is *I want a car* with the the expressive and evocative attitudes as described in Figure 8.3.

Following the generation algorithm introduced in Section 6.3.2, the realisation of the goal starts from NewInfo, and its relevance in the context is checked by the Relevance Criteria. The difference in the contexts is that the explicit attitudes in Context 8.2 concern needing a car while those in Context 8.3 concern wanting a car. The elliptical realisation of NewInfo (*Buy or rent?*) is Consistent in both contexts, but its Accuracy and Validity differs in them. The problematic concept is the Goal Concept `wantEvent(w, u, d)` which in Context 8.2 is implicit but in Context 8.3 explicit. In the former context, the concept `wantEvent(w, u, d)` belongs to the system’s private interpretation of the user’s explicitly expressed attitudes (the system has reasoned that

EXPRESSIVE ATTITUDES OF THE INPUT:

explicit want(u, know(s, [user(u), needEvent(n, u, c), car(c)]))

implicit want(u, know(s, [wantEvent(w, u, h), haveEvent(h, u, c), user(u), car(c)]))

EVOCATIVE ATTITUDES OF THE INPUT:

explicit want(u, want(s, know(s, [user(u), needEvent(n, u, c), car(c)]))),

implicit want(u, want(s, know(s, [wantEvent(w, u, h), haveEvent(h, u, c), user(u), car(c)])))

EVOKED ATTITUDES FOR THE RESPONSE:

know(s, [user(u), needEvent(n, u, c), car(c)]),

know(s, [wantEvent(w, u, h), haveEvent(h, u, c), user(u), car(c)]),

know(s, know(u, [wantEvent(w, u, d), car(c), user(u), hireEvent(r, u, c, -),

buyEvent(b, u, c, -), disj(d, b, r)]))

Figure 8.2: Expressive, evocative and evoked attitudes of the user contribution *I need a car*.

needing a car means wanting to have a car which for its part means wanting either to buy or rent a car, since these two alternatives are the only information that the system can find from its knowledge base to ‘explain’ the user need), while in the latter context it is part of the mutually known dialogue context. Ellipsis is thus considered Accurate and Valid in the latter context, since all the Goal Concepts (besides NewInfo) are mutually known (or inferable) in the immediate context and the user’s evocative attitudes have been appropriately addressed. However, in Context 8.2 ellipsis is considered Inaccurate and Invalid: the system cannot assume that this piece of information is implicitly communicated in an elliptical contribution, since the user may not be familiar with the system’s knowledge base and its ‘explanations’. To remedy Inaccuracy and Invalidity, the concept of the user’s want is added to the Agenda of the concepts to be realised. Syntactic constraints require that the object of the ‘buy/rent’ predicate is explicit in the surface contribution, and we assume that this actually leads to the generation of the full sentence *Do you want to buy or rent one?*¹³

Example (69) can be studied in the following contexts:

- At the beginning of the dialogue, the user contribution is *Are there any car hire*

¹³The pronoun “one” is preferred to “a car”, since this realises the Central Concept of the contribution.

EXPRESSIVE ATTITUDES OF THE INPUT:

explicit want(u, know(s, [user(u), wantEvent(w3, u, c3), car(c3)]))

implicit want(u, know(s, [wantEvent(w3, u, h3), haveEvent(h3, u, c3), user(u), car(c3)]))

EVOCATIVE ATTITUDES OF THE INPUT:

explicit want(u, want(s, know(s, [user(u), wantEvent(w3, u, c3), car(c3)]))),

implicit want(u, want(s, know(s, [wantEvent(w3, u, h3), haveEvent(h3, u, c3), user(u), car(c3)])))

EVOKED ATTITUDES FOR THE RESPONSE:

know(s, [user(u), wantEvent(w, u, c), car(c)]),

know(s, [wantEvent(w, u, h), haveEvent(h, u, c), user(u), car(c)]),

know(s, know(u, [wantEvent(w, u, d), car(c), user(u), hireEvent(r, u, c, -),

buyEvent(b, u, c, -), disj(d, b, r)]))

Figure 8.3: Expressive, evocative and evoked attitudes of the user contribution *I want a car*.

companies in Bolton? with the expressive and evocative attitudes as described in Figure 6.9, repeated here as Figure 8.4.

- At the beginning of the dialogue, the user contribution is *How many car hire companies are there in Bolton* with the the expressive and evocative attitudes as described in Figure 8.5.

EXPRESSIVE ATTITUDES OF THE INPUT:

want(u, know(u, [existq(co), carHireCompany(co), location(co, bolton)]))

EVOCATIVE ATTITUDES OF THE INPUT:

want(u, want(s, know(u, [existq(co), carHireCompany(co), location(co, bolton)])))

EVOKED ATTITUDES FOR THE RESPONSE:

know(s, [existq(co), carHireCompany(co), location(co, bolton)])

know(s, [cardinality(setOf(co, _cos), 12), carHireCompany(co), location(co, bolton)])

know(s, know(u, showall(12, carHireCompany(co))))

Figure 8.4: Expressive, evocative and evoked attitudes of the user contribution *Are there any car hire companies in Bolton?*

As in the previous example, the planning starts from NewInfo which is now the cardinality of the set of car hire companies in Bolton.¹⁴ In Context 8.5 its elliptical

¹⁴Since the number of companies exceeds a certain limit, set in our case to 5, the cardinality rather than the list of companies is considered NewInfo.

EXPRESSIVE ATTITUDES OF THE INPUT:

```
want(u, know(u,
  [cardinality(setOf(co, _cos), _C), carHireCompany(co), location(co, bolton)]))
```

EVOCATIVE ATTITUDES OF THE INPUT:

```
want(u, want(s, know(u,
  [cardinality(setOf(co, _cos), _C), carHireCompany(co), location(co, bolton)])))
```

EVOKED ATTITUDES FOR THE RESPONSE:

```
know(s, [cardinality(setOf(co, _cos), 12), carHireCompany(co), location(co, bolton)])
know(s, know(u, showall(13, carHireCompany(co))))
```

Figure 8.5: Expressive, evocative and evoked attitudes of the user contribution *How many car hire companies are there in Bolton?*

realisation as *12.* is Accurate, Consistent, Valid and Free From False Implicatures. However, in Context 8.4, the elliptical *12.* is not appropriate at all. The contribution is Consistent and Free From False Implicatures, but not Accurate (it does not express the existence of car hire companies), nor Valid (it does not address the user's evocative attitudes about the existence of car hire companies). To remedy this, the system has to express explicitly the attitude concerning the existence of car hire companies. This can be done in two ways: either the system makes the contribution Accurate by realising the 'existence' attitude in the surface contribution, or it addresses the user's evocative attitudes and makes the contribution Valid by providing an explicit answer to the user question. In the former case, the result is a full sentence¹⁵: *There are 12 car hire firms in Bolton.* which is also Valid. In the latter case, two elliptical expressions are produced: *Yes. 12.* and the contribution is Accurate as well. The order of the elliptical expressions is determined by the heuristics which says that explicit addressing of previous evocative attitudes comes first in the contribution. The choice between which criteria to fulfil first is based on contextual reasoning. We assume that the system attempts to fulfil Validity before Accuracy: it is more cooperative and considerate to ensure that the partner's evocative attitudes are fulfilled than to ensure that one's own goal is accurately conveyed. Failing to fulfil the user's evocative attitudes is more prone to convey unwanted implicatures than failing to fulfil Accuracy. The user is

¹⁵We assume that existence is explicitly expressed by the existential *there is/there are* construction.

not aware of the system's own goals, but if the system does not address the evocative attitudes of the user goal, there must be a reason for this; had the system understood the evocation, it would have provided the expected response.

If the response had been based on the realisation of the 'existence' attitude, the contribution *Yes* would have been Valid (it addresses the user's evocative attitudes), but not Accurate (NewInfo is not conveyed at all) nor Free From False Implicatures. It would have carried the wrong implicature that the system does not know how many car hire companies there are in Bolton (had the system known the cardinality, it would have come up with an answer that specifies this).

This example shows that the order in which the Relevance Criteria are applied is crucial to the surface realisation of the goal. Although the result must satisfy all the criteria simultaneously, the order of their satisfaction determines what kind of syntactic constructions are available. We consider this another important sign of the interaction between strategic and tactical planning and a support for our claim that communicative knowledge can be successfully used to bridge the generation gap.

The implicitness of existentiality here can be supported by the fact that an explicit confirmation marker *yes* is preferred to a simple existential sentence in the example below:

- (70) U: Do you know if there are any car hire companies in Bolton?
S10: Yes. There are 12 car hire companies in Bolton.
S11: *There are 12 car hire companies in Bolton.

The request for the number of car hire companies in Bolton is embedded syntactically deeper than in the original example: the question about existentiality is embedded in the yes-no question about the hearer's knowledge. The inference chain to reach the indirect request seems to be too long to warrant an explanatory answer without also providing an explicit response to either of the surface yes-no-questions. Notice that the confirmation is ambiguous with respect to which yes-no question it addresses ('Do you know' or 'Are there'), but this does not hinder the communication of relevant concepts. The ambiguity would only become problematic if the response were a simple 'yes'.

8.5 Conclusion

In this chapter we studied how the system plans the surface level representation of its responses. We assumed that the communicative goal of the system is given. The distinction between goal formulation and response planning concerns now the implicitness and explicitness of the communicative content: goal formulation determines the desirable next state of the dialogue and thus selects the concepts that must be explicitly or implicitly communicated, while response planning determines the concepts which are explicitly communicated to the partner. Both are based on communicative principles, and can effectively influence the surface realisation: the generation gap between content planning and realisation can thus be bridged.

Explicitness is operationalised via the ‘Responsiveness Principle’ and the ‘Minimalism Principle’, corollaries of ideal cooperation and rational agenthood. The communicative goal contains the focussed information which the system is to communicate explicitly or implicitly to the user. The response planner augments and organises these concepts into a response in such a way that it addresses the coherence and cooperativeness requirements of the current dialogue state. The planning process is guided by four relevance criteria which constrain cooperative and successful communication with the help of the speakers’ intentions and beliefs encoded in the expressive and evocative attitudes. Each concept that is explicitly expressed must be relevant according to the relevance criteria.

The motivation for the response is to exchange information, and thus the speaker needs to express explicitly only the information that the partner is not assumed to know. Repetition of old contextual information triggers an inference procedure concerning the motivation for giving such information.

Chapter 9

Implementation

In this chapter, we describe how the principles of CAA can be implemented in a dialogue system. We describe the implementation of a dialogue manager which can handle simple information seeking dialogues. Development was done in SICStus Prolog version 2.1, running under UNIXTM on a Sun Sparc Station SLC.

The dialogue manager is not fully robust in its interpretative capabilities, but we believe that the basic principles implemented in the prototype are also valid for full-scale dialogue systems.

9.1 Data structures

Crucial to any dialogue system explicitly based on pragmatic principles is the notion that the context influences both interpretation and the planning of responses. If we model the behaviour of the dialogue participant by a series of *processes* (from interpretation to response planning), we see the context as a shared memory which is both updated by and accessed by those processes. The memory contains more or less persistent data. This includes memory of past dialogue events (utterances) which persist until forgotten (a phenomenon that we greatly simplify here). It also includes information which is subject to change. For example, the Central Concept is subject to

constant change, and the representation of the speakers' mental states such as beliefs and intentions may change as a consequence of dialogue events. The most obvious way to model the context in Prolog is to implement it as data structures in the assertional database which persist until explicitly retracted.

The implementation is based on context updates and the dynamic knowledge base called Context Model is the basis of reasoning. The Context Model consists of the following data structures (explained in Chapter 6).

Contribution:

User and system contributions are recorded in a predicate `contr/3` whose arguments describe the 'overt' properties of the contribution. The first argument is the sequential number of the contribution in the course of the dialogue, the second one encodes the speaker as the constant *u* (user) or *s* (system), and the third argument encodes the input sentence as a string of words:

```
contr(ContrNumber,Speaker,Sentence).
```

Goal:

The communicative goal of a contribution is stored as a three-place predicate `goal/3`. The first argument refers to the contribution whose goal is in question, the second argument is the owner of the goal (the speaker of the contribution), and the third argument is a list of desirable states. We assume that each contribution is associated with exactly one communicative goal, and if a goal consists of several desirable states, these are rhetorically related to each other and support the same goal:

```
goal(ContrNumber,Speaker,GoalStates).  
goal(2,s,[know(s,[wantEvent(w,u,h),user(u),car(c),hireEvent(r,u,c,_),  
buyEvent(b,u,c,_),disj(d,b,r)]))]).
```

Unfulfilled Goals:

The system has its own ‘private goals’, formulated on the basis of its role as an information provider. If such a goal is not fulfilled, either because the system has followed a related user initiative or because the user has deliberately left the system initiative unaddressed, the unfulfilled goal is stacked for possible future fulfilment. The predicate `unfulfilledGoals/2` encodes an unfulfilled goal. The first argument is the contribution which introduced the goal, and the second argument is the goal which needs to be fulfilled in the course of the dialogue.

```
unfulfilledGoals(ContrNumber,UnfulfilledGoal).
```

Expressive, evocative and evoked attitudes:

The system entertains three sets of attitudes: expressive, evocative and evoked, referring to the attitudes that the speaker expresses, intends to evoke in the hearer, and presupposes, respectively. These are encoded as two-place predicates `ex/2`, `ec/2` and `ev/2`, respectively. The first argument in each predicate refers to the contribution they belong to, and the second argument is an attitude:

```
ex(ContrNumber,Attitude).  
ec(ContrNumber,Attitude).  
ev(ContrNumber,Attitude).
```

Each attitude is encoded in the attitude language, briefly described below. The attitude consists of the attitude operator, the participant whose attitude is in question, and the content. The attitude operator is either ‘want’, ‘know’ or ‘know-ref’, referring to intention and beliefs about the truth-value of a proposition and value for a referent, respectively. The participant is either ‘s’ or ‘u’, depending on whether the attitude is held by the system or the user, respectively. The content is either another attitude (nested attitudes), or a list of world-model concepts. The examples below give expressive attitudes for the user wanting the system to know that the user needs a car and the system wanting to know the location of a hire-event, evocative attitudes for the user

wanting the system to want to know that the user needs a car, and the system wanting the user to want the system to know the location of a hire-event, and finally, the evoked attitude for the system knowing that the user knows the location of a hire-event.

```
ex(1,want(u, know(s, [needEvent(n,u,c), user(u), car(c)]))) .
ex(4,want(s, know-ref(s, [location(h,_), hireEvent(h,u,c,_), user(u),
car(c)]))) .
```

```
ec(1,want(u, want(s, know(s, [needEvent(n,u,c), user(u), car(c)])))) .
ec(4,want(s, want(u, know-ref(s, [location(h,_), hireEvent(h,u,c,_),
user(u), car(c)])))) .
```

```
ev(4, know(s, know-ref(u, [location(h,_), hireEvent(h,u,c,_), user(u),
car(c)]))) .
```

Discourse referents:

Discourse referents are instantiated world model concepts that have been introduced in the course of the dialogue. They are produced from the content of the expressed attitudes, and they are encoded in the one-place predicates `dr/1` whose only argument is the instantiated world model concept. The argument is the whole concept and not just the index of a concept:

```
dr(hireEvent(h,u,c,_)) .
dr(car(c)) .
```

Central Concept:

Each contribution has a Central Concept which is a discourse referent being currently talked about. Central Concept is encoded as a two-place predicate¹ `topic/2` whose first argument refers to the contribution in question, and the second argument is a discourse referent (not just the index of the discourse referent, but the whole concept):

```
topic(ContrNumber, CentralConcept) .

topic(2, car(c)) .
topic(4, hireEvent(h,u,c,_)) .
```

¹In earlier development of the work, Central Concept was called ‘topic’.

NewInfo:

The new information conveyed in each contribution is encoded in the predicate `new/2`. A contribution may contain several concepts that are new information, and these are encoded as separate predicates with the same first argument. Most often `NewInfo` is a single concept.

```
new(ContrNumber,NewInfo).
```

```
new(1,needEvent(n,u,c)).  
new(1,user(u)).  
new(1,car(c)).  
new(2,disj(d,b,r)).
```

InfoGathered:

The information which has been gathered for a database search is encoded in the predicate `infoGathered/2`. The first argument is in a database query format as a table with slots filled with the information gathered so far, and the second argument is a list of world model concept that specify the query:

```
infoGathered(Query,Specs).
```

```
infoGathered([carHireCompany,_,_,_,bolton,_],  
             [existq(co),location(co,bolton)]).
```

9.2 Control structure

The implementation is based on the task division given in Figure 9.1. Dashed boxes refer to the natural language front end.

The interaction with the user consists of three tasks (processes, phases): accept an input, interpret the input and react to the input. Acceptance deals with the evaluation of contact and perception: reading and parsing the input. Interpretation, the details of which we simplify here, is divided into two subtasks: getting the user goal

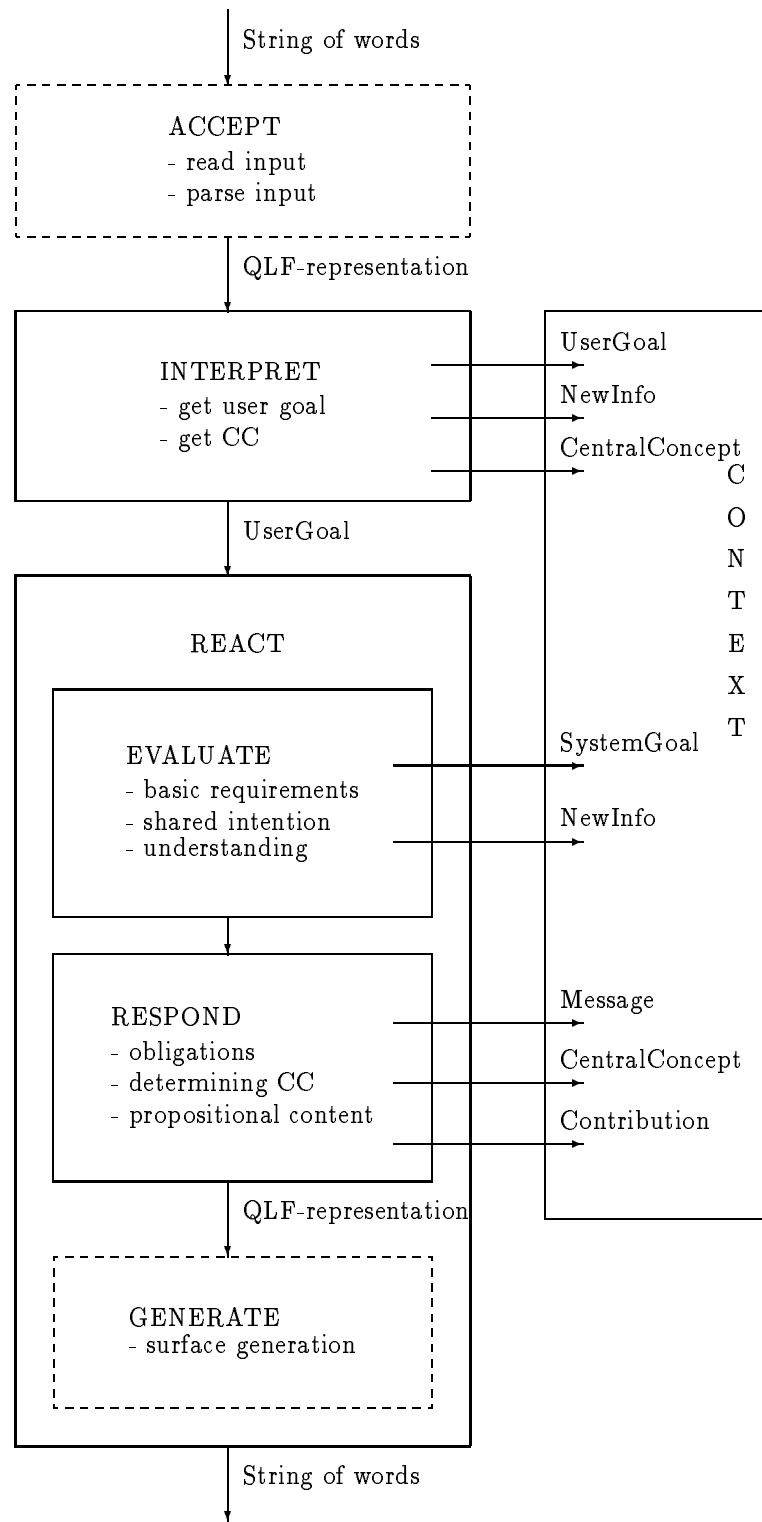


Figure 9.1: Data flow in the system.

and determining the thematic coherence of the contribution, i.e. Central Concept and NewInfo. Reaction includes evaluation of, and response to, the user goal. Evaluation is divided into three subtasks: basic requirements (the check whether the system is willing and able to continue and perceive contributions), shared intention (the system reasons about the joint purpose), and understanding (the system formulates its own goal on the basis of joint purpose, world knowledge and application knowledge). Responding consists of three tasks: complying with the system obligations, determining Central Concept, and planning the surface level representation of the response. The QLF-representation is output to the surface generator which produces the appropriate string of words.

The program is started by calling the predicate `st/0` which is as follows:

```
st :-
    systemContribution0,
    initialise!,          /* from initial.pl */
    caa!,
    again.
```

The initial situation is such that there is contact but no perception of user input, and the system's greeting contribution has taken place. The initialisation of the context is done by `initialise/0` by asserting the following facts in the knowledge base:

<code>contact(t).</code>	contact exists
<code>perception(nil).</code>	no perception
<code>contr(0,sys,'Welcome to EYP. Please state your query.').</code>	initial system contribution
<code>goal(0,s,[know(u,_)])</code>	system goal is to provide information to the user
<code>dr(system(s)).</code>	system is the only discourse referent
<code>infoGathered([]).</code>	no information gathered for database search
<code>ex(0,want(s, know(u, _)))</code>	system has expressed that it wants the user to know something (information about YP)
<code>ec(0,want(s,want(u, know(u, _))))</code>	system wants to evoke the user's want to know something (information about YP)
<code>topic(nil,nil).</code>	no previous Central Concept
<code>topic(0,greet).</code>	Central Concept is to greet
<code>new(0,greet).</code>	new information is to greet

The main predicate is `caa/0` which is as follows:

```
caa :-
    accept(N,Input),!,
    interpret(N,Input),!,
    react(N),!,           /* from rp.pl */
    caa.                  /* expect a user response */
```

The predicates `interpret/2` and `react/1` correspond to the dialogue manager. The natural language front-end is embedded in the predicates `accept/2` and `react/1`, see below.

When the response has been given to the user, `caa/0` is called again, thus enabling a dialogue between the system and the user. The dialogue can be terminated by the user by explicit closing phrase (*Bye*).

9.3 Acceptance of the user contribution

The predicate `accept/2` takes care of establishing the contact and perception, asserting the user contribution with the input sentence, and parsing. It delivers the QLF-representation of the input sentence.

```
accept(N,Parsed) :-
    contact(t),
    retract(perception(_)),
    read(Input),
    contrNumb(N),
    asserta(contr(N,user,Input)),
    parser(N,Input,Parsed),
    nl,write('Parse: '),nl,write(Parsed),nl,nl.
```

Contact and perception are handled by the two predicates `contact/1` and `perception/1`. Their single argument is 't' or 'nil' depending on whether the system has contact with the user and is able to perceive her contributions. Most often these basic requirements of communication are fulfilled once the user has started the program. However, these predicates are not only 'theoretical sugar'. If the user has not provided any input for

a certain length of time, the system can conclude that there is no contact any more and close the connection (cf. the discussion in Section 2.3.2 on the problema of closing in the collected dialogue corpus). This can be done via calls to UNIX routines. The system's ability to 'perceive' can be understood as its capability to parse (see below).

9.3.1 Parsing

We assume an existing Categorical Grammar-based Prolog parser, described in Lager and Black (1994). The system's 'perception' is defined on the basis of the parser being able to produce a QLF-representation of the surface string of words. If the parser can parse the input sentence, perception is flagged 'ok'. If the parser cannot produce any parse, perception is flagged 'nil', and the evaluation process results in the goal 'beAble' (system wants to be able to parse).

```

parser(N,In,Sem) :-
    parse(In),                                /* from cky_ccg_parser.p */
    parse_result(_:Sem),!.                    /* from cky_ccg_parser.p */
    assert(perception(t)),!.
parser(_,_,[]) :-
    assert(perception(nil)),!.

```

The parser parses the input string into a quasi-logical form. The quasi-logical form is a conjunction of semantic predicates, used in the PLUS project and further developed later. The representation is described in Jokinen (1993a). Parser results for some possible user inputs are listed below.

```

parser(_,'I need a car',
    [i(_u),num(_u,sg),case(_u,nom),def(_u),need(_n,_u,_c),
     stype(_n,finf),tense(_n,pres-time),num(_n,sg),pers(_n,st),
     forsome(_c),car(_c),num(_c,sg),case(_c,acc),kind(_c,countn)]).

parser(_,'Rent',
    [rent(_r,_1,_2),stype(_r,finf),tense(_r,pres-time),num(_r,_3),
     pers(_r,_4)]).

parser(_,'In Bolton',
    [in(_i,_2,_b),pname(_b,bolton),def(_b),num(_b,sg),case(_b,acc)]).

```

```

parser(_, 'Are there any car hire companies in Bolton',
  [interrog(_1), existq(_c), forsome(_c), carhirecompany(_c),
   num(_c, pl), case(_c, nom), kind(_c, countn), in(_i, _c, _b),
   pp-def(_i), pname(_b, bolton), def(_b), num(_b, sg), case(_b, acc)]).

parser(_, 'How many car hire companies are there in Bolton',
  [interrog(_1), how_many(_c), carhirecompany(_c),
   num(_c, pl), case(_c, nom), kind(_c, countn), existq(_c),
   stype(_1, finf), tense(_1, pres-time), num(_1, pl), pers(_1, rd),
   in(_i, _c, _b), pp(_i), pname(_b, bolton), def(_b), num(_b, sg),
   case(_b, acc)]).

parser(_, 'Are there any vegetarian restaurants in Manchester',
  [interrog(_1), existq(_r), forsome(_r), vegetarian(_2, _r),
   basef(_2), restaurant(_r), num(_r, pl), case(_r, nom),
   kind(_r, countn), in(_i, _r, _m), pp-def(_i), pname(_m, manchester),
   def(_m), num(_m, sg), case(_m, acc)]).

parser(_, 'Is the Tandoori Kitchen an Indian restaurant',
  [interrog(_1), be(_1), stype(_1, finf), tense(_1, pres-time),
   num(_1, sg), pers(_1, rd), pname(_tk, 'Tandoori Kitchen'),
   def(_tk), num(_tk, sg), case(_tk, nom), forsome(_tk),
   indian(_2, _tk), basef(_2), restaurant(_tk), num(_tk, sg),
   case(_tk, nom), kind(_tk, countn)]).

parser(_, 'What Chinese restaurants are there in Rusholme',
  [interrog(_1), what(_r), chinese(_2, _r), basef(_2),
   restaurant(_r), num(_r, pl), case(_r, nom), kind(_r, countn),
   existq(_r), stype(_1, finf), tense(_1, pres-time), num(_1, pl),
   pers(_1, rd), in(_i, _1, _ru), pp(_i), pname(_ru, rusholme),
   def(_ru), num(_ru, sg), case(_ru, acc)]).

parser(_, 'I need food',
  [i(_u), num(_u, sg), case(_u, nom), def(_u), need(_n, _u, _f),
   stype(_n, finf), tense(_n, pres-time), num(_n, sg), pers(_n, st),
   food(_f), num(_f, sg), case(_f, acc), kind(_f, massn)]).

parser(_, 'I want a car',
  [i(_u), num(_u, sg), case(_u, nom), def(_u), want(_w, _u, _c),
   stype(_w, finf), tense(_w, pres-time), num(_w, sg), pers(_n, st),
   forsome(_c), car(_c), num(_c, sg), case(_c, acc), kind(_c, countn)]).

parse(_, 'What types of restaurants do you list',
  [interrog(_1), what(_t), type(_t), num(_t, pl), case(_t, nom),
   kind(_t, countn), of(_o, _t, _r), restaurant(_r), num(_r, pl),
   case(_r, acc), kind(_r, countn), you(_s), num(_s, nd), case(_s, nom),
   list(_l, _s, _r), stype(_l, finf), tense(_l, pres-time), num(_l, sg),
   pers(_l, nd)]).

```

```

parse(_, 'List all car hire companies',
      [imper(_1), list(_1, _s, _c), stype(_1, imper), tense(_1, pres-time),
       num(_1, sg), pers(_1, nd), all(_c), carhirecompany(_c), num(_c, pl),
       case(_c, acc), kind(_c, countn)]).

```

9.4 Interpretation of the user contribution

The thesis concentrates on dialogue management and planning of system responses, and thus interpretation of user contributions is not implemented in its full form. The interpretation is handled by the predicate `interpret/2`. It takes the contribution number and QLF-representation as its input and updates the contextual knowledge base with the user goal, the expressive and evocative attitudes associated with the goal and with Central Concept and NewInfo of the contribution. The predicate has no output variables: its operation is shown in the knowledge base updates.

The predicate `interpret/2` contains three predicates: `translate/3`, `userGoal/3` and `userObligations/3`. The first one translates the semantic representation into WM concepts, the second one takes care of the interpretation of the user contribution with respect to the Context Model, world model and communicative principles, and the last one checks that user obligations are fulfilled and updates the context accordingly. Since the work has concentrated on the planning and generation side, the predicate `interpret/2` is currently only partially implemented in the system.

```

interpret(N, Parsed) :-
    translate(Parsed, Concepts, Mood),
    userGoal(N, Concepts, Goal),
    userObligations(N, Goal, Mood).

```

9.4.1 Translation of the semantic representation

The QLF-representation is translated into world model concepts via a bidirectional conceptual lexicon. The predicate `translate/3` deals with this task:


```

translate(Parsed,Concepts,Mood) :-
    mood(Parsed,Mood,Pars),
    qlfToConc(Pars,Concepts).

```

The predicate `mood/3` distinguishes the surface communicative act (statement, question, request) on the basis of the sentence mood: declarative, interrogative, and imperative. Sentence mood is separated from the rest of the parser output, and used later in asserting the user-related obligations into the context.

The predicate `qlfToConc/2` does the translation from QLF-predicates into world model concepts. The mappings between QLF-predicates and world model concepts are specified in the conceptual lexicon which can be used both in interpretation and generation. There are three types of QLF-predicates: content-word predicates, auxiliary predicates and ignored predicates. Content-word predicates correspond to WM concepts which can occur as discourse referents, auxiliary predicates are translated into auxiliary concepts which specify the goal, but do not contribute to the population of the discourse world, and ignored predicates are left out of the conceptual representation because these are assumed to have no impact on the reasoning. Auxiliary semantic predicates are e.g. those related to there-constructions (*are_there*, *is_there*), cardinality (*how_many*), proper names and different restaurant types and they correspond to concepts ‘existq’, ‘cardinality’, ‘pname’ and ‘typeof’, respectively. Ignored QLF-predicates are for instance ‘tense’, ‘def’, ‘case’ and ‘pp-def’. Of course, what is ignored depends on the structure of the world model and the reasoning it is designed to perform – our world model mainly reasons about events, objects and their relations.

```

qlfToConc([],[]).
qlfToConc([Lex|Lexs],[Conc|Concs]) :-
    cl(Lex,Conc),!,                               /* from cl.pl */
    qlfToConc(Lexs,Concs).
qlfToConc([Lex|Lexs],Concs) :-
    ignore(Lex),!,                               /* from cl.pl */
    qlfToConc(Lexs,Concs).
qlfToConc([Lex|Lexs],[unknown(Lex)|Conc]) :-
    qlfToConc(Lexs,Conc).

```

A part of the the small conceptual lexicon is shown below:

```

/*      event(evVar,Subj,Obj,Company)      */
cl(rent(_e,_x,_y,_p),hireEvent(_e,_x,_y,_p)).
cl(buy(_e,_x,_y),buyEvent(_e,_x,_y,_z,_p)).
cl(sell(_e,_x,_y),buyEvent(_e,_z,_y,_x,_p)).
cl(steal(_e,_x,_y),stealEvent(_e,_x,_y,_p)).

/*      object(objVar)                      */
cl(company(_x),companies(_x)).
cl(car(_x),car(_x)).
cl(bike(_x),bike(_x)).
cl(restaurant(_x),restaurant(_x)).

/*      restaurant type adjectives */
cl(chinese(_x,_y),typeof(_y,chinese)).
cl(french(_x,_y),typeof(_y,french)).
cl(indian(_x,_y),typeof(_y,indian)).
cl(italian(_x,_y),typeof(_y,italian)).

/*      specs needed in db-search */
cl(how_many(_c),cardinality(setOf(_c,_Cs),_s)).
cl(existq(_c),existq(_c)).
cl(pname(_p,_s),pname(_p,_s)).

/*      ignore these in the WM translation */
ignore(case(_,_)).
ignore(pers(_,_)).
ignore(num(_,_)).
ignore(tense(_,_)).

```

9.4.2 Recognition of the user goal

The user's communicative goal is a list of belief states that represent the desired next state in the dialogue. The user may also have other goals that the system is not aware of. For instance, the user may be planning a trip to Europe and thus needs to know about car hire possibilities. However, the role of a simple information provider does not require the system to understand any other user goals except those related to requesting and receiving information on a particular topic. The task structure is thus very simple, and the communicative goals become the goals that the system attempts to recognise.

The implementation of the predicate `userGoal/4` is beyond the chosen scope of the

thesis. It takes care of the recognition of the user goal and NewInfo, determination of the Central Concept and the context updates. Input to userGoal/4 is contribution number, grammatical mood and the translated concepts, and output is the user goal. An approximation to its definition used in our implementation is given below.

```
userGoal(N,Mood,Concepts,Goal) :-
    goalRecognition(N,Mood,Concepts,Goal,New),!,
    assertGoal(N,u,Goal,New),
    assertDrs(Concepts),
    nl,write('User Goal: '),nl,write(Goal),nl,nl,
    ccFilterA(N,Concepts,Goal,CC),!,    /* from topic.pl */
    nl,write('Central Concept: '),write(CC),nl,
    write('NewInfo: '),write(New),nl,nl.
```

The following list summarises the output of goalRecognition/5 for some user inputs. The actual user input is given in *italics*, and the three lists refer to the corresponding WM concepts, the assumed user goal and the NewInfo of the contribution. The content of the goal is expressed as a list of world model concepts.

I need a car:

```
goalRecognition(_,decl,
[user(u),needEvent(n,u,c),car(c)],
[know(s,[wantEvent(w,u,h),haveEvent(h,u,c),user(u),car(c)])],
[user(u),needEvent(n,u,c),car(c)]).
```

Rent: (after the question 'Do you want to buy or rent a car?')

```
goalRecognition(_,decl,
[hireEvent(r,u,c,_)],
[know(s,[wantEvent(w,u,r),hireEvent(r,u,c,_),user(u),car(c)])],
[hireEvent(r,u,c,_)]).
```

In Bolton: (after the question 'Where do you want to rent a car?')

```
goalRecognition(_,decl,
[location(r,b),pname(b,bolton)],
[know(s,[location(r,bolton),hireEvent(r,u,c,_),user(u),car(c)])],
[location(r,bolton)]).
```

Are there any car hire companies in Bolton?:

```
goalRecognition(_,interrog,
[existq(co),carHireCompany(co),location(co,b),pname(b,bolton)],
[know(u,[existq(co),carHireCompany(co),location(co,bolton)])],
[existq(co)]).
```

How many car hire companies are there in Bolton?:

```
goalRecognition(_,interrog,
[cardinality(setOf(co,_cos),_C),carHireCompany(co),existq(co),
  location(co,b),pname(b,bolton)],
[know(u,[cardinality(setOf(co,_cos),_C),carHireCompany(co),
  location(co,bolton)])],
[cardinality(setOf(co,_cos),_C)]).
```

Are there any vegetarian restaurants in Manchester:

```
goalRecognition(_,interrog,
[existq(re),typeof(re,vegetarian),restaurant(re),location(re,m),
  pname(m,manchester)],
[know(u,[existq(re),typeof(re,vegetarian),restaurant(re),
  location(re,manchester)])],
[existq(re)]).
```

Is Tandoori Kitchen an Indian restaurant?:

```
goalRecognition(_,interrog,
[pname(tk,'Tandoori Kitchen'),typeof(tk,indian),restaurant(tk)],
[know(u,[pname(tk,'Tandoori Kitchen'),typeof(tk,indian),restaurant(tk)])],
[typeof(tk,indian)]).
```

What Chinese restaurants are there in Rusholme?:

```
goalRecognition(_,interrog,
[typeof(x,chinese),restaurant(x),existq(x),location(x,ru),
  pname(ru,rusholme)],
[know(u,[typeof(x,chinese),restaurant(x),existq(x),location(x,rusholme)])],
[typeof(x,chinese)]).
```

What Indian restaurants are there in Rusholme?:

```
goalRecognition(_,interrog,
[typeof(xi,indian),restaurant(xi),existq(xi),location(xi,ru),
  pname(ru,rusholme)],
[know(u,[typeof(xi,indian),restaurant(xi),existq(xi),location(xi,rusholme)])],
[typeof(xi,indian)]).
```

I want a car:

```
goalRecognition(_,decl,
[user(u),wantEvent(w3,u,c3),car(c3)],
[know(s,[wantEvent(w3,u,h3),haveEvent(h3,u,c3),user(u),car(c3)])],
[user(u),wantEvent(w3,u,c3),car(c3)]).
```

What types of restaurants do you list:

```
goalRecognition(_,interrog,
[whattypes(rs,_),restaurant(rs),system(s),listEvent(l,s,rs)],
[know(u,[whattypes(rs,_),restaurant(rs)])],
[whattypes(rs,_)]).
```

List all carhirecompanies:

```
goalRecognition(_,imper,
[listEvent(l2,s,co2),listall(co2,_),carHireCompany(co2)],
[do(s,[listall(co2,_),carHireCompany(co2)]]),
[listall(co2,_)])
```

After inferring the user goal, the Central Concept is abduced from the user goal and the existing discourse referents. The predicate `ccFilterA/4` deals with the CC determination, and it is shown below together with the expected output for some user inputs. The three first arguments of the predicate are input arguments corresponding to the contribution number, the list of surface concepts, user goal and the last argument is the output, the Central Concept.

```
ccFilterA(N,SurfConc,Goal,CC) :-
    ccCandidatesRec(N,SurfConc,Goal,Candidates),
    prevTopic(N,Pcc),
    possibleCC(Pcc,Candidates,Shifts),!,
    nl,write('Possible user shifts: '),nl,write(Shifts),nl,nl,
    chooseBest(N,Shifts,SurfConc,CC),!,
    assertz(topic(N,CC)).
```

I need a car:

```
ccFilterA(_,
    [user(u),needEvent(n,u,c),car(c)],
    [knows(s,[wantEvent(w,u,h),haveEvent(h,u,c),user(u),car(c)])],
    needEvent(n,u,c)).
```

Rent: (after the question 'Do you want to buy or rent a car?')

```
ccFilterA(_,
    [hireEvent(r,u,c,_)],
    [knows(s,[wantEvent(w,u,r),hireEvent(r,u,c,_),user(u),car(c)])],
    car(c)).
```

In Bolton: (after the question 'Where would you like to rent a car?')

```
ccFilterA(_,
    [location(r,b),pname(b,bolton)],
    [knows(s,[location(r,bolton),hireEvent(r,u,c,_),user(u),car(c)])],
    hireEvent(r,u,c,_)).
```

Are there any car hire companies in Bolton?:

```
ccFilterA(_,
    [existq(co),carHireCompany(co),location(co,b),pname(b,bolton)],
    [knows(u,[exists(co),carHireCompany(co),location(co,bolton)])],
    carHireCompany(co)).
```

How many car hire companies are there in Bolton?:

```
ccFilterA(_,
    [cardinality(setOf(co,_cos),_C),carHireCompany(co),
     existq(co),location(co,b),pname(b,bolton)],
    [knows(u,[cardinality(setOf(co,_cos),_C),
              carHireCompany(co),location(co,bolton)]]),
     carHireCompany(co)).
```

Are there any vegetarian restaurants in Manchester:

```
ccFilterA(_,
    [existq(re),typeof(re,vegetarian),restaurant(re),
     location(re,m),pname(m,manchester)],
    [knows(u,[existq(re),typeof(re,vegetarian),restaurant(re),
              location(re,manchester)]]),
     restaurant(re)).
```

Is Tandoori Kitchen an Indian restaurant?:

```
ccFilterA(_,
    [pname(tk,'Tandoori Kitchen'),typeof(tk,indian),restaurant(tk)],
    [know(u,[pname(tk,'Tandoori Kitchen'),typeof(tk,indian),
             restaurant(tk)])],
     restaurant(tk)).
```

What Chinese restaurants are there in Rusholme?:

```
ccFilterA(_,
    [typeof(x,chinese),restaurant(x),existq(x),location(x,ru),
     pname(ru,rusholme)],
    [know(u,[typeof(x,chinese),restaurant(x),existq(x),
             location(x,rusholme)]]),
     restaurant(x)).
```

What Indian restaurants are there in Rusholme?:

```
ccFilterA(_,
    [typeof(xi,indian),restaurant(xi),existq(xi),location(xi,ru),
     pname(ru,rusholme)],
    [know(u,[typeof(xi,indian),restaurant(xi),existq(xi),
             location(xi,rusholme)]]),
     restaurant(xi)).
```

I want a car:

```
ccFilterA(_,
    [user(u),wantEvent(w3,u,c3),car(c3)],
    [know(s,[wantEvent(w3,u,h3),haveEvent(h3,u,c3),user(u),car(c3)])],
     wantEvent(w3,u,c3)).
```

What types of restaurants do you list:

```
ccFilterA(_,
    [whattypes(rs,_),restaurant(rs),system(s),listEvent(l,s,rs)],
    [know(u,[whattypes(rs,_),restaurant(rs)])],
    restaurant(rs)).
```

List all carhirecompanies:

```
ccFilterA(_,
    [listEvent(l2,s,co2),listall(co2,_),carHireCompany(co2)],
    [do(s,[listall(co2,_),carHireCompany(co2)])]
    carHireCompany(co2)).
```

9.4.3 User obligations

Lastly, the system checks if the user obligations are fulfilled. The system ‘trusts’ that the user is a rational, cooperative agent and that the contribution fulfils the obligations. There are three types of obligations: sincerity (the user expresses and intends to evoke true attitudes), motivation (the user has reason to her contribution which the system can relate to its world model) and consideration (the user has consideration of the system’s ability to handle evocative intentions). The predicate `userObligations/3` takes care of checking the obligations.

```
userObligations(N,Concepts,Goal) :-
    sincere(N,Concepts),
    motivated(N,Goal),
    considerate(N,Goal).
```

9.5 Reaction to the user contribution

Our system takes a new pragmatics-based approach to generation, which attempts to overcome the boundary problems between planning and realisation. Given the communicative goal of the system in terms of desirable next states of the dialogue, the planner proceeds by refining the goal with respect to the contextual requirements and communicative obligations. The context and communicative principles influence not only the goal determination (what is communicated to the user), but also how this

knowledge is communicated to the user. We thus bridge the gap between the traditionally distinct what-to-say and how-to-say components (Meteor (1991) calls this the ‘generation gap’), by gradual specification of the system message and interleaving goal formulation and response planning.

Figure 9.2 shows the control flow in the system’s reaction to the user contribution. The evaluation and response modules form the system’s planner and generator.

The predicate `react/1` controls the planning and surface generation of the system response. Its only argument is the sequence number of the user contribution. It calls the predicate `evaluate/3` which deals with the evaluation of the user contribution with respect to the system’s cooperativeness principles, world knowledge and application domain, then updates the Context Model with the system goal and `NewInfo`, and finally calls `respond/2` which takes care of the fulfilment of system obligations and planning oriented towards surface realisation.

```
react(N) :-
    evaluate(N,SGoal,New),          /* from jointP.pl */
    nl,write('Sys Goal: '),nl,writeNumberedLines(SGoal,0),nl,nl,
    write('NewInfo: '),nl,write(New),nl,nl,
    M is N + 1,
    assertGoal(M,s,SGoal,New),!,
    respond(M,SGoal),!.             /* from rp.pl */
```

9.5.1 Evaluation of the user contribution

The requirements for ideal cooperative and full-blown communication deal with contact, perception, understanding and trust that the partner behaves so that the requirements are fulfilled. The partner’s contribution is evaluated on each level and the result of the evaluation is formulated as the system’s response or incorporated in it as feedback. We have simplified the response planning process and left out feedback mechanisms in our implementation.

The evaluation is dealt with by the predicate `evaluate/3` which consists of three predicates, `basicRequirements/1`, `expectations/3` and `understanding/4`.

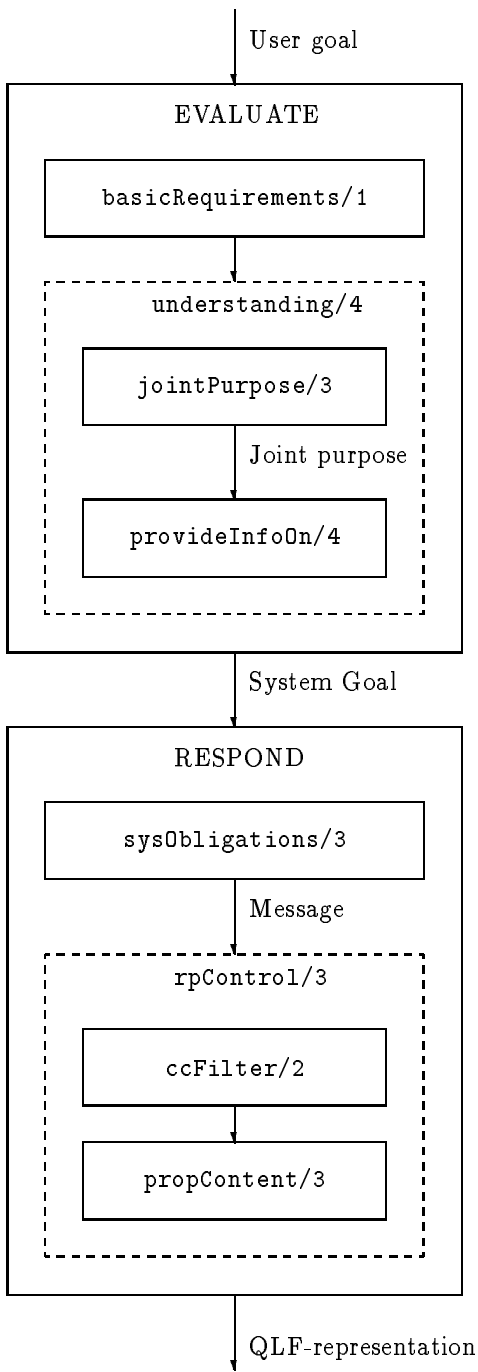


Figure 9.2: The control flow of the system's reaction.

```

evaluate(N,SGoal,New) :-
    basicRequirements(Flag),
    expectations(N,Flag,Shared),
    understanding(N,Shared,SGoal,New),!.

```

Basic requirements

Contact and perception, the two basic requirements for full-blown communication, are modelled by the predicate `basicRequirements/1` which checks the system's willingness to continue the contact and whether the system was able to perceive the contribution. They deal with the the evaluation of whether the basic requirements can still be considered valid and thus differ from the basic requirements checked at the beginning of the dialogue by the predicate `accept/2`: we are now dealing with the receiver's cooperativeness related to its *willingness* to continue contact and perception, and not with the receiver's *ability* to contact and perceive at all as before. Willingness is tied to the roles and power relations that the participants occupy in the dialogue, whereas ability has to do with the initial setting in which the communication takes place.

The system is assumed to be always 'willing' to continue contact, because of its role as an obedient servant and a cooperative information provider. If the system occupied another role in regard to the user, it could, of course, have more freedom in this respect, e.g. it could be allowed to quit the dialogue.² The system is also always 'willing' to perceive the user input and 'willing' to recover from perception failures. The system's 'ability' to perceive (whether it can parse the input sentence or not) is related to the system's 'willingness' to perceive, and used in the process of determining the joint purpose. If the parser was successful in parsing, the predicate `perception/1` is flagged 'true', and the system continues with the evaluation. However, if the parser failed, `perception/1` is flagged 'beAble', and this information is passed onwards to the predicate `understanding/3`, which formulates the appropriate goal, to be realised as a request to repeat the query.

²The system could, for instance, refuse to give information that is secret or confidential, or, on a different level, it could recognise swear words and decide not to continue with a user who uses unsuitable language.

```

basicRequirements(Flag) :-
    willingToContact(t),
    willingToPerceive(Flag),!.
basicRequirements(Flag) :-
    willingToContact(Flag),!.

willingToContact(t) :- contact(t).
willingToContact(quit).
willingToPerceive(t) :- perception(t).
willingToPerceive(beAble).

```

Expectations

If the basic requirements are fulfilled ($\text{Flag} = t$), evaluation proceeds to check whether there exists a shared intention in terms of communicative expectations set by the system earlier in the dialogue. The predicate `expectations/3` captures this: it checks whether the user contribution is in accordance with the system's stereotypical expectations about the dialogue continuation. If the system has asked a question, it expects the user to provide an answer, but if the system itself has given an answer, then it expects the user to initiate a new question. The predicate could be elaborated by adding heuristics about non-stereotypical expectations related to the chaining of different speech act types. However, as argued earlier, we aim to use world model knowledge, the dialogue context and participants' attitudes in reasoning about the relevance of a contribution, and do not model speech acts as such.

```

expectations(N,t,expect) :-                                /* expected responses */
    M is N - 1,
    ( ec(M,want(s,want(u, know(s,_))))),                  /* ask(sys),      */
    ec(N,want(u,want(s, know(s,_))))),                  /* answer(user) */
    ;
    ec(M,want(s,want(u, know(u,_))))),                  /* answer(sys),   */
    ec(N,want(u,want(s, know(u,_))))),                  /* ask(user)      */
    ;
    ec(M,want(s,want(u, know(u,_))))),                  /* answer(sys),   */
    ec(N,want(u,want(s, do(s,_))))),                    /* request(user) */
    ),!.
expectations(_,t,unexpect).                                /* unexpected responses */
expectations(_,Flag,Flag).                                 /* failed basic requirements */

```

Understanding

The predicate `understanding/4` deals with understanding the user goal, and can be seen as starting the traditional ‘strategic planning’, or planning the actual content of the system response. It consists of two tasks: reasoning about the joint purpose and filtering this purpose via the application model (and providing information from the database if any relevant information is found). Its inputs are the contribution number and the information about the expectation fulfilment and its outputs are the system goal and `NewInfo`. Depending on whether the user response fulfils the communicative expectations or not, the system determines the joint purpose by using the predicate `jointPurpose/3` or `jointPurposes/3`, respectively.

```
understanding(_,repeated,[know(u,alreadyGiven)], [alreadyGiven]) :-
    nl,write('Joint purpose: '),nl,write(alreadyGiven),nl,nl.
understanding(N,expect,Goal,New) :-      /* expected responses */
    jointPurpose(N,Purpose,F),
    nl,write('Joint purpose: '),nl,write(Purpose),nl,nl,
    provideInfo(F,Purpose,Goal,New),!.
understanding(N,unexpected,Goal,New) :-   /* unexpected responses */
    jointPurposes(N,Purpose,F),
    nl,write('Joint purpose: '),nl,write(Purpose),nl,nl,
    provideInfo(F,Purpose,Goal,New),!.
understanding(_,Goal,[Goal],[Goal]) :-   /* failed basic requirements */
    nl,write('Joint purpose: '),nl,write(Goal),nl,nl.
```

Joint purpose

The predicates `jointPurpose/3` and `jointPurposes/3` deal with the system’s cooperativeness and rationality in regard to joint activity. They encode how to react to the user goal on the dialogue level: what kind of response to provide, and how to resolve conflicts if the user goal is different from the system goal. Both of them take the contribution number as input, and they output the joint purpose as a list of world model concepts as well as a flag which shows whether the user contribution has been mutually understood ($F = true$), contains a world model misconception ($F = mis$), or is incomprehensible with regard to the system’s world knowledge ($F = nil$). The flag is passed over to the predicate `provideInfo/4`, and the appropriate action is taken

depending on the flag.

The following control flow shows how the joint purpose is determined:

```
while userGoal
  if contact
    then if perception
      then if understood
        then if not misconception
          then if jointPurpose
            then jointP = provideInfoOn(userGoal)
            else jointP = clarify;notrelated;
              somethingElse;repeat
          else jointP = toCorrect
        else jointP = toUnderstand
      else jointP = beAble
    else jointP = quit
```

The joint purposes **quit** and **beAble** result from the basic requirements of **contact** and **perception** having failed. The first of these is not currently used in our system because of the system's role as a cooperative always-willing information provider. However, the goal is included for the sake of completeness. The joint purpose **toUnderstand** results if the user contribution does not make sense in the system's conceptual world, i.e. the system has not been able to understand the user with respect to its world model and application model. The joint purpose **toCorrect** results if the system has detected a misconception in the user's contribution. However, because of the lack of a comprehensive world model, the implemented system does not deal with the user's misconceptions or their correction. If needed, something like McCoy's ROMPER system (McCoy, 1984) could be plugged in here to specify the joint purpose **toCorrect**, or the ideas in Joshi et al. (1984) could be used.

If the basic requirements are fulfilled and the user contribution is mutually understood, joint purpose is determined on the basis of four parameters discussed in Section 5.2.3: the system checks if the expectations are fulfilled, if the user contribution is related in content to what the previous dialogue was about, if the system has its own goals to pursue, and if the partner has the initiative. The following algorithms encode the

system's decisions on how to continue the dialogue and what kind of joint purpose to choose. The corresponding predicate is listed as well.

1. Algorithm for the expected user contributions (expressive attitudes of the user contribution are evocative attitudes of the previous system contribution):

```

If expected
  then if related
    then if sysinit
      then revert
      else followup
    else if sysinit
      then repeatNew
      else if unfulfilledGoals
        then newquestion
        else newrequest

```

Gloss: If the user's reaction is the one that was intended to be evoked, then the system's alternatives are: if the user contribution is related to the dialogue context, then the system reverts to a previous unfulfilled goal or continues with the follow-up user goal depending on whether the system has the initiative or not; if the user contribution is not related to the dialogue context and the system has the initiative, then the system repeats the NewInfo of the last goal, otherwise if the system has unfulfilled goals, it assumes that the user asked an unrelated question, otherwise it assumes that a totally new request starts.

Predicate:

```

jointPurpose(N,JPurpose,true) :-
    related(N,true),           /* related contribution */
    ( initiative(N,sys),       /* system initiative */
      backto(N,JPurpose),!       /* system goes back to previous goal */
    ;
      followup(N,JPurpose),!     /* user asked a related */
    ;
      followup(N,JPurpose),!     /* followup question */
    ).

jointPurpose(N,JPurpose,true) :- /* unrelated contribution */
    ( initiative(N,sys),       /* user gave unrelated response */
      unfulfilledGoals,       /* system has goals */
      repeatNew(N,JPurpose),!   /* repeat NewInfo of the last goal */
    ;
      unfulfilledGoals,       /* system has goals */
      newquestion(N,JPurpose),! /* user asked unrelated question */
    ;
      newrequest(N,JPurpose),!  /* user asked totally new request */
    ).

```

2. Algorithm for the unexpected user contributions (expressive attitudes of the user contribution are not evocative attitudes of the previous system contribution):

```

If unexpected
then if related
    then if sysinit
        then subquestion
        else if unfulfilledGoals
            then revert
            else somethingElse
    else if sysinit
        then repeatCC
        else if unfulfilledGoals
            then notrelated
            else newrequest

```

Gloss: If the evoked user reaction is not the one that was intended to be evoked, then the system has the following alternatives: if the user contribution is related to the dialogue context and the system has the initiative, then the system continues with the user goal which is a clarification subquestion, otherwise it reverts to a previous unfulfilled goal or assumes that the user has given feedback and wants to know something else, depending on whether the system has unfulfilled goals; if the user contribution is not related to the dialogue context and the system has the initiative, then the system is persistent and repeats its previous goal, otherwise it assumes that the user question was not related or that a totally new request starts.

Predicate:

```

jointPurposes(N,JPurpose,true) :-
    related(N,true),                /* related contribution */
    ( initiative(N,sys),            /* system has initiative */
      subquestion(N,JPurpose),!       /* user starts subdialogue */
    ;
      unfulfilledGoals,            /* system has goals */
      backto(N,JPurpose),!
    ;
      somethingElse(N,JPurpose),!    /* user has given feedback */
    ).

jointPurposes(N,JPurpose,true) :-    /* unrelated contribution */
    ( initiative(N,sys),            /* user response was unrelated and */
      unfulfilledGoals,            /* unexpected */
      repeatCC(N,JPurpose),!         /* repeat CC of the last goal */
    ;
      unfulfilledGoals,            /* user question was unrelated and */
      notrelated(N,JPurpose),!       /* unexpected */
    ;
      newrequest(N,JPurpose),!       /* user stated totally new request */
    ).

```

Relatedness is tested with the help of Central Concept: the contributions must talk

about the same topic. The Central Concept constraint could be extended to check plans, tasks, rhetorical relations etc. to determine if the utterance is relevant and coherently related to the previous dialogue. However, in our implementation, we simply assume that the contribution is related to the previous dialogue if it is related to the previous CC.

Providing information for the system response

When the joint purpose has been determined, this is passed on to the predicate `provideInfo/4`. Depending on the flag, the joint purpose is filtered via the application model, or one of the goals `toCorrect` and `toUnderstand` is formulated. It is important to notice that the user contributions are interpreted first with respect to the world knowledge, and only after this reasoning filtered via the application knowledge. Hence user contributions are treated as being related to the entire world knowledge rather than just to the knowledge in the application.

```
provideInfo(true,JPurpose,SGoal,New) :-
    understand(JPurpose,JPurpose2),
    provideInfoOn(JPurpose2,SGoal,[New]).
provideInfo(mis,_,[toCorrect],[toCorrect]).
provideInfo(nil,_,[toUnderstand],[toUnderstand]).
```

The predicate `provideInfoOn/3` deals with application model filtering and formulating the system goal to the user. It takes the list of joint purpose concepts as its input and outputs the system goal together with the associated `NewInfo`. In some cases, such as when the system wants to know if the user wants some other information (*somethingElse*) or when the system wants to repeat the previous goal (*repeat*), the goal is construed without reference to the system's application knowledge. In other cases, the message is built on the basis of the system's metaknowledge of the domain (*whattypes*) or by consulting the actual database (*askYP*).

```
provideInfoOn([somethingElse],[know(s,somethingElse)],somethingElse).

provideInfoOn([repeat(X)|Goal],[know(u,repeat(X))|Goal],repeat).
```



```

provideInfoOn([clarify(Unclear)],SGoal,New) :-
    present(Unclear,SGoal,New),!.

provideInfoOn([newquest(Top)|Concepts],[know(s,newquest(Top))|SGoal],New) :-
    mapWmToAm(Concepts,AMHeadings,Specs),!,
    nl,write('AM headings: '),write(AMHeadings),nl,nl,
    ( AMHeadings = [],
      YPInfo = [],
      write('No headings'),nl,nl,!
    );
    tooManyHeadings(AMHeadings,Concepts,YPInfo),
    write('Too many headings'),nl,nl,!
    ;
    amInfo(AMHeadings,Specs,YPInfo),! /* from ypm.pl */
),
present2(Concepts,YPInfo,SGoal,New).

provideInfoOn([listall(P,_)|Q],SGoal,New) :-
    mapWmToAm([listall(P,_)|Q],Heading,Specs1),!,
    delAll(listall(P,_),Specs1,Specs2),
    wholeList(Heading,Specs2,YPInfo), /* from ypm.pl */
    present([listall(P,_)|YPInfo],SGoal,New).

provideInfoOn([whattypes(Cl,_)|Rest],SGoal,New) :-
    mapWmToAm([whattypes(Cl,_)|Rest],[Heading],_),!,
    askYP(whattypes(Heading),[],Info),!, /* from ypm.pl */
    present([whattypes(Cl,_)|Info],SGoal,New).

provideInfoOn([backto(Top),Info,Specs],[know(u,backto(Top))|SGoal],New) :-
    askYP(Info,Specs,YPInfo),!, /* from ypm.pl */
    present2(Top,YPInfo,SGoal,New).

provideInfoOn(Purpose,SGoal,New) :-
    mapWmToAm(Purpose,AMHeadings,Specs),!,
    nl,write('AM headings: '),write(AMHeadings),nl,nl,
    ( AMHeadings = [],
      YPInfo = [],
      write('No headings'),nl,nl,!
    );
    tooManyHeadings(AMHeadings,Purpose,YPInfo),
    write('Too many headings'),nl,nl,!
    ;
    amInfo(AMHeadings,Specs,YPInfo),! /* from ypm.pl */
),
present2(Purpose,YPInfo,SGoal,New).

```

Mapping of joint purpose to application domain

The joint purpose, expressed as a set of world model concepts, must be related to the application model so that a database query can be formulated. In our case, the system's application model is a small database concerning Yellow Pages information about restaurants and car hire companies in the Manchester area, and the application model concepts are Yellow Pages headings. The joint purpose is mapped onto the headings by the predicate `mapWmToAm/3`. If only one applicable heading is found, the database is queried by the predicate `amInfo/3`, otherwise an appropriate message and the set of headings are returned.

Most event-type and object-type concepts are mapped to YP-headings, and a concept may be mapped onto several YP-headings if the concept is associated with several service suppliers. On the other hand, not all concepts are related to a YP-heading: usually only the 'basic level' concepts are included in the mapping relation, and the superconcepts are excluded. For instance, the concepts 'car' and 'bike' are mapped to particular headings, but their superconcept 'vehicle' has no YP-relation. In a similar way, the concepts 'needEvent' and 'stealEvent' have no corresponding YP-headings: the former is too general and the latter refers to an illegal event, and thus they are not associated with any particular service supplier.

A number of auxiliary concepts are included in the mapping as well. For example, 'location' and 'typeof' (restaurant type) serve as constraints on the database search. However, they cannot initiate a database query alone, since they are not mapped to any YP-headings.

The following predicates provide the mapping between the world model and the application model (the indexing of the Yellow Pages headings to world model concepts is supported both ways for quick reference):

- ypSupplier/2 maps WM event concepts to YP headings:

```
ypSupplier(stealEvent(_,_,_,_), []).
ypSupplier(hireEvent(_,_,_,_), [carHireCompany, flatAgency, bikeShop]).
ypSupplier(buyEvent(_,_,_,_),
           [carGarage, restaurant, grocery, shoeShop, bikeShop]).
```

- amRelated/2 maps WM object concepts to YP headings:

```
amRelated(car(_), [carHireCompany, carGarage]).
amRelated(restaurant(_), [restaurant]).
amRelated(company(_), [carHireCompany, flatAgency, bikeShop]).
```

- specs/1 lists WM concepts which are not mapped to YP headings, but are needed in the database search:

```
specs(cardinality(_,_)).
specs(location(_,_)).
specs(typeof(_,_)).
```

- enables/2 maps YP headings to WM event concepts:

```
enables(carHireCompany, hireEvent(_,_,_,_)).
enables(carGarage, buyEvent(_,_,_,_)).
enables(carGarage, repairEvent(_,_,_,_)).
enables(restaurant, buyEvent(_,_,_,_)).
enables(restaurant, eatEvent(_,_,_,_)).
```

- involves/2 maps YP headings to WM object concepts:

```
involves(carHireCompany, [car(_), van(_), company(_), carHireCompany(_)]).
involves(carGarage, [car(_), van(_)]).
involves(restaurant, [restaurant(_), food(_)]).
```

It is assumed that the query is reasonable with respect to the application model. Impossible queries like the one originating from the contribution *I repair food* are ruled out before the mapWmToAm/3 predicate is even invoked: inconsistent goals are discarded on the basis of World Model reasoning by the predicate jointPurpose/3 (misconception).

Database access

Application model headings and the specifications are the input for the predicate `amInfo/3` which produces the information from the database. It first checks if there already exists some information gathered for the database search and then formulates a YP-query using `formulateQuery/4` or `formulateYPQuery/3` depending on whether such information exists or not, respectively. If there is already information gathered for a database search, the query formulation checks if this information is consistent with the current request, and if not, it delivers a notice of the inconsistency (such a situation would be e.g. when the system has gathered information about Indian restaurants in Rusholme, but the current request deals with Chinese restaurants). If the query is consistent with the gathered information, the YP is queried and the result of this query delivered.

```
amInfo(AMHeadings,Specs,Info) :-
    infoGathered(Query,_),
    formulateQuery(AMHeadings,Specs,Query,Notice),
    ( Notice = ok,
      nl,write('Query: '),write(Query),nl,nl,
      nl,write('Specs: '),write(Specs),nl,nl,
      askYP(Query,Specs,YPInfo),!,
      retract(infoGathered(_,_)),
      asserta(infoGathered(Query,Specs)),
      Info = YPInfo,!
    ;
      Info = [Notice],! ).
amInfo(AMHeadings,Specs,YPInfo) :-
    formulateYPQuery(AMHeadings,Specs,Query),
    nl,write('Query: '),write(Query),nl,nl,
    nl,write('Specs: '),write(Specs),nl,nl,
    askYP(Query,Specs,YPInfo),!,
    asserta(infoGathered(Query,Specs)).
```

The YP-query is in the form of a list `[C1,T,Na,Ad,L,Te]` which contains the following information:

Cl = class or the YP-heading: restaurant, carhire company

T = type of: only for restaurants (italian, french, vegetarian), carGarage (sale, repair) and bikeShop (sale, repair)

Na = name of the company

Ad = address of the company

L = location of the company: manchester, manchester 14, bolton, manchester airport

Te = telephone number of the company

Only single queries are dealt with; queries like *Restaurants in loc1 and loc2* or *Restaurants or groceries in loc* cannot be answered.

From application model to system goal

The system goal is formulated by present2/4 which presents YP-information as a set of belief states. One of the beliefs is the main evocative intention, in our case this corresponds to the first belief in the list. The other goal attitudes are supporting ones, and they are realised if needed (cf. RST type nucleus and satellites, (Mann & Thompson, 1987)). The main system goals are as follows:

- If no headings were available for the database search, the system wants to understand the request:

```
present([], [toUnderstand], [[toUnderstand]]).
```

- If the user's request is ambiguous with respect to the application model (too many headings), a clarification question about the headings is asked, and NewInfo is instantiated as the disjunct.

```
present([headings, Headings, Purpose],  
        [know(s, [wantEvent(w, u, d) | Info])], New) :-  
    presentInfo(Headings, Purpose, Info, New).
```

- If the result of the database query is nil, the system wants the user to know this. NewInfo is the cardinality nil. Compensation concerning the information which caused the database search failure (class was not found, types were not found, etc.) is planned later by sysObligations/3.

```

present([number(nil),Class|Specs],
        [know(u,[cardinality(setOf(In,[]),nil),Class|Specs])],
        [[cardinality(setOf(In,[]),nil)])] :-
        Class =.. [_ ,In].

```

- If the number of companies found is less than some constant, in our case 5, the system wants the user to know the list. NewInfo is the YP information.

```

present([number(Num),Specs|YPInfo],
        [know(u,supplyInfo(Specs)),
         know(u,[cardinality(setOf(In,YPInfo),Num)|Specs]),
         know(u,ypinfo(In,YPInfo))],
        [[ypinfo(In,YPInfo)])] :-
        Specs = [Class|_],
        Class =.. [_ ,In].

```

- If the number is more than 5, but there is a YP-based criterion to limit the number, the system wants to know how to specify the criterion and optionally, inform the user of the cardinality of the found set as well. NewInfo concerns the specification.

```

present([ask(Spec,Num),Key|YPInfo],
        [know(s,[Specs,Class]),
         know(u,[cardinality(setOf(In,YPInfo),Num)|Key])], [New]) :-
        Key = [Class|_],
        dr(Class),
        Class =.. [Cl,In],
        presentSpec(Cl,In,Spec,Specs,New).

```

- If the number is more than 5, and there is no YP-based criterion to limit the number, the system wants the user to know the companies and their total number and also wants to know whether the user wants to see the whole list. NewInfo is the number, the request whether to show all the YP information and the YP information itself.

```

present([whole(Num),Specs|YPInfo],
        [know(u,[cardinality(setOf(In,YPInfo),Num)|Specs]),
         know(s,showall(S,Class)),
         know(u,ypinfo(In,YPInfo))],
        [[cardinality(setOf(In,YPInfo),Num),ypinfo(In,YPInfo)])] :-
        gensym(1,S),
        Specs = [Class|_],
        Class =.. [_ ,In].

```

- If the request has concerned the types in the database, the system wants to inform the user of the type list and the number of different types. NewInfo is the type list.

```

present([whattypes(In,_),types(Cl,Num),TypeList],
        [know(u,[whattypes(In,TypeList),Class]),
         know(u,[cardinality(setOf(types(Cl),TypeList),Num),Class])]),
        [[whattypes(In,TypeList)]] ) :-
        Class =.. [Cl,In].

```

- Finally, if the attempted database search is inconsistent, because the information previously gathered does not match the information in the current query, the inconsistent piece of information is presented to the user. This is also the NewInfo.

```

present([inc(X,0,Cl)], [know(u,inc(X,0,Cl))], [[inc(X,0,Cl)]]).

```

Consolidating the system goal

The evaluation results in a system goal which describes the desired next state of the dialogue. The final task in the evaluation process is to update the Context Model with the system's communicative goal as well as the new information to be communicated to the user. There are two different ways of encoding the next state: we can describe the state as a set of beliefs or use simple directives that abbreviate a particular set of attitudes and refer to a particular type of response to be given. We prefer the first alternative and represent the goal as a set of beliefs. This is in accordance with the gradual specification of the goal: the belief set is not realised as such on the surface level, but modified by the response planner so that the communicative obligations are fulfilled.

However, we also have four special goals which deal with the communication of overall management information, as opposed to the communication of some piece of information from the system's knowledge base: *quit*, *beAble*, *toUnderstand* and *toCorrect*. The distinction is reflected e.g. in the encoding of the goals: management goals are simple directives which will be directly translated into a surface string. The planning of a contribution from the content goals requires reference to the obligations of sincerity, motivation and attention, but in management goals the obligations are 'precompiled' into the goal itself.

```
goal(4,beAble).
goal(2,[knows(s,[wantEvent(w,u,d),disj(d,b,r),buyEvent(b,u,c),
             hireEvent(h,u,c),user(u),car(c)])))]).
```

9.5.2 Responding to the user contribution

The goal is expressed as a set of *beliefs* that the speaker wants to make true. The system *intentions* are formed from the beliefs when the system obligations are checked: the planner selects which facts to realise according to the dialogue context.

The system's reaction is completed by formulating a surface level response. This is done by calling the predicate `respond/2` which calls the predicate `rpGo/3` to build the semantic representation for the system message and the predicate `sgenerate/2` to produce the actual surface string.

```
respond(N,SGoal) :-
    write('----- Entering Response Planner'),
    rpGo(N,SGoal,SemRepr),!,
    write('----- Leaving Response Planner'),nl,nl,
    write('The semantic representation is:'),nl,nl,
    showSemRepr(SemRepr),nl,nl,
    sgenerate(SemRepr,SurfString),
    asserta(contr(N,sys,SurfString)),
    ( retract(explicit(_))
      ;
      true ).
```

The predicate `rpGo/3` is needed to distinguish between management goals and other goals, since the latter need to be specified with respect to the communicative obligations while the obligations are precompiled into the former. In the case of management goals, `rpGo/3` updates the context and produces the semantic representation directly, while the other goals go through the obligation checking and the propositional content is then built for this obliged goal, or the message by `rpController/4`.


```

rpGo(N,[quit],[[quit]]) :-
    assertAttitudes(N,s,u,[know(u,[wantEvent(_,s,quit)])]),
    retract(contact(t)),
    assert(contact(nil)),
    assertz(topic(N,quit)),!.
rpGo(N,[beAble],[[cannot_parse]]) :-
    assertAttitudes(N,s,u,[know(u,[cannot_parse])]),
    assertAttitudes(N,s,u,[do(u,[rephrase])]),
    assertz(topic(N,rephrase)),nl,!.
rpGo(N,[toUnderstand],[[not_understand]]) :-
    assertAttitudes(N,s,u,[know(u,[not_understand])]),
    assertAttitudes(N,s,u,[do(u,[repeat])]),
    assertz(topic(N,not_understand)),nl,!.
rpGo(N,[toCorrect],[[incorrect]]) :-
    assertAttitudes(N,s,u,[know(u,[incorrect])]),
    assertAttitudes(N,s,u,[do(u,[repeat])]),
    assertz(topic(N,incorrect)),nl,!.
rpGo(N,Goal,SemRepr) :-
    nl,write('_____ Checking system obligations'),nl,nl,
    sysObligations(N,Goal,Message),
    nl,write('Obliged goal: '),
    nl,writeNumberedLines(Message,0),nl,nl,!,
    rpController(N,Message,SemRepr,1).

```

9.5.3 The system's obligations

The system's competence as a cooperative rational agent is encoded in the communicative obligations which constrain the set of beliefs that the system wants to make true. The obligations are thus not only restrictions on how the desired goal state can be best obtained or presented to the user, but also restrictions on what is a reasonable desired next state. Thus content planning and realisation interact and are interleaved in the planning of system responses.

The obligations are sincerity, motivation and consideration. Sincerity refers to the speaker's truthfulness: the speaker is sincere in that she intends to give true information or she believes that the partner can provide the missing piece of true information. Motivation concerns the speaker's grounding of the goal: the speaker can support what she is communicating by having evidence for what she intends to say. Motivation can be compared to the discussion on 'justifying why' in Webber and Joshi (1982): the

system needs to be able to explain why it has given a particular response.

The obligation of consideration is sometimes difficult to distinguish from motivation: the reason why something is said can be based on considerations about the partner's needs and knowledge. However, there is an important distinction between these two types of obligations: motivation deals with the speaker's commitment to what she knows and wants, while consideration is related to her commitment to what she believes the partner knows and wants.

Below we list some implementations of the communicative obligations.

Motivation: the speaker can support the response.

1. Everything that the system wants to know is motivated. In our system, the reason to ask comes from four sources: the system is unable to perform a database search (too many headings), it needs information to restrict the search (ask specification), it did not understand the user's question (rephrase), and it could not parse the input, i.e. failed to perceive the input (repeat).

```
motivation(N,[know(s,P)|R],[know(s,P)|Q]) :- motivation(N,R,Q).
```

2. Everything that the user wanted to know or wanted the system to do is motivated.

```
motivation(N,[know(u,P)|R],[know(u,P)|Q]) :-  
    M is N - 1,  
    ( ec(M,want(u,want(s,know(u,P1)))) ,  
      contains(P1,P)  
    ;  
      ec(M,want(u,want(s,do(s,P1)))) ,  
      contains(P1,P) ),  
    motivation(N,R,Q).
```

3. Everything that informs about inconsistency is motivated.

```
motivation(_,[know(u,inc(X,Old,C1))],[know(u,inc(X,Old,C1))]).
```

4. Informing about the repetition of the same request is motivated.

```
motivation(_,[know(u,alreadyGiven)],[know(u,alreadyGiven)]).
```

5. If the response was unrelated or the system wants to repeat the goal, it is motivated to inform the user about the irrelevance.

```
motivation(_,[know(u,notrelated(CC))],[know(u,notrelevant(CC))]).  
motivation(_,[know(u,repeat(X))|Goal],[know(u,notrelevant(X))|Goal]).
```

6. If the user did not specifically ask cardinality, it is motivated by the joint purposes 'newquest', 'supplyInfo' and the intention to know whether the user wants to see all service suppliers.

```

motivation(N,[know(s,newquest(T))|R],
           [know(s,newquest(T)),know(u,[cardinality(S,K)|Spec])|Q])
:-
  member(know(u,[cardinality(S,K)|Spec]),R),
  delAll(know(u,[cardinality(S,K)|Spec]),R,R1),
  motivation(N,R1,Q).
motivation(N,[know(u,supplyInfo(Sp))|R],
           [know(u,supplyInfo(Sp)),know(u,[cardinality(S,K)|Spec])|Q])
:-
  member(know(u,[cardinality(S,K)|Spec]),R),
  delAll(know(u,[cardinality(S,K)|Spec]),R,R1),
  motivation(N,R1,Q).
motivation(N,[know(u,[cardinality(S,K)|Spec])|R],
           [know(u,[cardinality(S,K)|Spec]),know(s,showall(I,X))|Q])
:-
  member(know(s,showall(I,X)),R),
  S = setOf(C,_),
  X =.. [_ ,C|_],
  delAll(know(s,showall(I,X)),R,R1),
  motivation(N,R1,Q).
motivation(N,[know(u,[cardinality(S,nil)|Spec])|R],
           [know(u,[cardinality(S,nil)|Spec])|Q]) :-
  motivation(N,R,Q).

```

Consideration: the speaker attends to the partner's need as a rational cooperative agent: all goal beliefs are considered simultaneously.

1. If cardinality is nil, add compensation on the basis of goal attitudes (P) and cardinality specification (Rest).

```

consideration(_,P,[know(u,[cardinality(Class,nil)|Rest]),Compensation])
:-
  member(know(u,[cardinality(Class,nil)|Rest]),P),!,
  compensation(P,Rest,Compensation).

```

2. If NewInfo is not relevant and the system has the initiative, inform of irrelevance and repeat the goal.

```

consideration(_, [know(u,notrelevant(new(_)))|Goal],
               [know(u,notrelevant)|Goal]).

```

3. If CC is not relevant and the system has the initiative, inform of irrelevance, give explanation and repeat the goal.

```

consideration(_, [know(u,notrelevant(topic(CC)))|Goal],
               [know(u,notrelevant),know(u,talkingAbout(CC))|Goal]).

```

4. If CC is not relevant and the user has the initiative, inform of irrelevance and ask if the user wants to continue with previous topic.

```

consideration(_, [know(u, notrelevant(CC))],
               [know(u, notrelevant),
                know(s, [wantEvent(ws, u, g), continueEvent(ce, u, T), user(u), CC])])
:-
    CC =.. [_ , T | _].

```

5. If the system wants to go back to previous CC, the previous goal is repeated. A 'backto'-marker is added if the CC was discussed more than three contributions earlier.

```

consideration(N, [know(u, backto(CC)) | G], P) :-
    topic(M, CC),
    ( N - M > 3,
      P = [know(u, backto(CC)) | G]
    ;
      P = [G] ).

```

6. If the joint purpose is newquest and the application model filtering has resulted in the system wanting to know something, then the system wants to know how to proceed as follows: if the filtering results in the system wanting to know if the user wants to see all the companies, then consideration deals with the system wanting the user to know the cardinality and the system wanting to know whether the user wants to see the companies (continue with her request) or continue with the previous CC; if the filtering results in the system wanting to know a specification parameter for the database search, then consideration deals with the system wanting the user to know the cardinality and the system wanting to know whether the user wants to specify her request or continue with the previous CC; if the filtering results in too many headings, the consideration deals with the system wanting to know whether the user wants to continue with her request or with the previous Central Concept.

```

consideration(N, [know(s, newquest(CC)) | R], P) :-
    member(know(s, X), R),
    ( X =.. [showall, In, _], /* want to see all companies */
      member(know(u, [cardinality(setOf(Co, Cos), Num) | Sp]), R),
      P = [know(u, [cardinality(setOf(Co, Cos), Num) | Sp]),
           know(s, [wantEvent(W, u, D), disj(D, In, C), X,
              continueEvent(C, u, T), user(u), CC])])
    ;
      X =.. [What, _ | []], /* specify location or type */
      member(know(u, [cardinality(setOf(Co, Cos), Num) | Sp]), R),
      gensym(1, S),
      P = [know(u, [cardinality(setOf(Co, Cos), Num) | Sp]),
           know(s, [wantEvent(W, u, D), disj(D, S, C),
              specifyEvent(S, u, What),
              continueEvent(C, u, T), user(u), CC])])
    ;

```

```

        topic(N,Topic),          /* clarify toomanyheadings */
        Topic =.. [_ ,In|_],
        gensym(1,C1),
        P = [know(s,[wantEvent(W,u,D),disj(D,C1,C),
                                continueEvent(C1,u,In),Topic,
                                continueEvent(C,u,T),user(u),CC]])]
    ),
    CC =.. [_ ,T|_],
    gensym(1,W),
    gensym(1,D),
    gensym(1,C).

```

7. If the joint purpose is newquest and the application model filtering has resulted in the system wanting the user to know YPInfo, then the system wants the user to know the YPInfo and also the system wants to know if the user wants to continue with previous Central Concept.

```

consideration(_, [know(s,newquest(CC)),know(u,[YPInfo])],
    [know(u,[YPInfo]),
        know(s,[wantEvent(W,u,C),continueEvent(C,u,T),user(u),CC])]) :-
    CC =.. [_ ,T|_],
    gensym(1,W),
    gensym(1,C).

```

Sincerity: the speaker is truthful.

```

sincerity(_,[],[]).
sincerity(N,[know(s,P)|Rest],Rest2) :-          /* user has previously */
    ec(_,want(u,want(s,know(s,P)))) ,           /* given the info          */
    sincerity(N,Rest,Rest2).
sincerity(N,[know(s,P)|Rest],[know(s,P)|Rest2]) :-
    assertz(ev(N,know(s,know(u,P)))),
    sincerity(N,Rest,Rest2).
sincerity(N,[know(u,P)|Rest],[know(u,alreadyGiven)|Rest2]) :-
    ec(_,want(s,want(u,know(u,Q)))) ,           /* system has previously */
    ground(Q),P = Q,                             /* given the info          */
    sincerity(N,Rest,Rest2).
sincerity(N,[know(u,P)|Rest],[know(u,P)|Rest2]) :-
    assertz(ev(N,know(s,P))),
    sincerity(N,Rest,Rest2).
sincerity(N,[do(s,_)|Rest],[do(s,_)|Rest2]) :-
    sincerity(N,Rest,Rest2).
sincerity(N,[R|Rest],[R|Rest2]) :-
    sincerity(N,Rest,Rest2).

```

9.5.4 Response planning

The ‘obliged goal’, or the system’s message may contain several beliefs that need to be realised in the same contribution. `rpController/4` feeds them to `rpControl/3` which takes one communicative intention at a time and forms a semantic representation for it. This step requires a rather strong assumption about the content of the message: each communicative intention is assumed to correspond to a clause-like chunk whose relevance and relation to other intentions is specified implicitly by its inclusion into the message during the checking of obligations. It should be noticed that intentions could also be grouped together into one surface construction, cf. argumentation in Horacek (1992), but we have not included this facility in the prototype system. We also assume that one of the goal intentions is the main intention that determines the expectations that the system will have about the next user contribution.

`rpControl/3` first determines the view point from which the `NewInfo` is presented to guarantee the thematic coherence of the surface string. The Central Concept is determined by the predicate `ccFilter/2`. The propositional content for the surface contribution is built on the basis of `NewInfo` and Central Concept by the predicate `propCont/3`.

```
rpController(_, [], [], _).
rpController(N, [Goal|Goals], [Sr|SemRepr], Num) :-
    nl, write('Goal '), write(Num), write(' to be realised:'),
    nl, write(Goal), nl, nl, Num2 is Num + 1,
    rpControl(N, Goal, Sr), !,
    rpController(N, Goals, SemRepr, Num2).

rpControl(N, Goal, SemRepr) :-
    nl, write('----- Determining Central Concept'), nl, nl,
    ccFilter(N, Goal), !, /* from topic.pl */
    nl, write('----- Determining Propositional Content'), nl, nl,
    propContent(N, Goal, SemRepr), !.
```

(For system development purposes, the planner can also be run separately. It is then started with the predicate `rp/0` which first lets the user select the goal and the context, then initialises the context according to the user selection, and finally calls `rpGo/3`).

Central Concept determination

The Central Concept is determined by the predicate `ccFilter/2`. It functions according to the following algorithm: given the contribution number and the goal, the possible next CCs for the goal are first predicted (`ccCandidates/3`), then the shifts from the previous CC to the CC-candidates are formulated with the help of domain-based coherence relations (`possibleCCs/3`), and finally the best alternative is selected according to communicative principles (`selectBest/3`).

```
ccFilter(Number,Goal) :-
    prevTopic(Number,Pcc),
    ccCandidates(Number,Goal,Candidates),
    ( possibleCCs(Pcc,Candidates,Shifts),!,
      nl,write('Possible system shifts: '),nl,write(Shifts),nl,nl,
      selectBest(Number,Shifts,CC),!,
      nl,write('Central Concept: '),write(CC),nl,nl,
      assertz(topic(Number,CC))
    ;
      nl,nl,
      write('No coherent continuation available for the previous CC: '),
      write(Pcc),nl,
      write('Cannot abduce next CC from the goal concepts: '),
      nl,nl,write(Candidates),
      nl,nl,!,fail ).
```

A possible CC is thematically related to the previous dialogue. This means that the shift from the previous CC to a CC candidate is coherent. A coherent shift is defined as follows:

1. no shift at all (CC continues),
2. the shift conforms to the topic shift rules,
3. the shift is awkward, but is explicitly marked with a Topic-shift marker.

For a given previous CC, coherent shifts must be restricted to those that are relevant in the dialogue context. The set of goal concepts serves as the constraint for search space: only shifts to CC candidates are checked.

Central Concept candidates are goal concepts which are not NewInfo or the dialogue participants (system(s), user(u)). Preferably the candidates are current discourse referents but if no such concepts are available, then accessible discourse referents are tried. Accessible discourse referents are possible topics from some previous CC, i.e. they are thematically related to some previous CC. If no accessible discourse referents are available either, a totally new view-point is introduced in the dialogue. However, this kind of topic shifting is not possible in our prototype, which will fail in this case.

Coherent shifts are listed below (cf. Section 7.2.3):

- From an event to the next action in sequence or sub/superaction:

```
coherentShift(Pcc,CC) :-
    type(Pcc,events),
    partOfPlan(Pcc,CC).          /* from wm.pl */
```

- From an event to the participants of the event:

```
coherentShift(Pcc,CC) :-
    type(Pcc,events),
    isArgument(CC,Pcc),          /* from wm.pl */
    typeCheck(CC,Pcc).
```

- From an event to the location of the event. There are two kinds of settings in the world model: companies and locations. Companies occur as arguments of actions (e.g. where the object is bought/hired from), locations refer to proper locations like 'Manchester':

```
coherentShift(Pcc,CC) :-
    type(Pcc,events),
    type(CC,locations),
    locCheck(CC,Pcc).
```

- From an object to the event in which the object participates:

```
coherentShift(Pcc,CC) :-
    type(Pcc,objects),
    isArgument(Pcc,CC),          /* from wm.pl */
    typeCheck(Pcc,CC).
```

- From an object to the attributes of the object:


```

coherentShift(Pcc,CC) :-
    type(Pcc,objects),
    attribute(CC,Pcc).          /* from wm.pl */

```

- From an attribute to the object whose attribute this is:

```

coherentShift(Pcc,CC) :-
    type(CC,objects),
    attribute(Pcc,CC).          /* from wm.pl */

```

- From a location to the event which takes place in the location:

```

coherentShift(Pcc,CC) :-
    type(Pcc,locations),
    type(CC,events),
    locCheck(Pcc,CC).

```

- From location to sub- or super-location:

```

coherentShift(Pcc,CC) :-
    type(Pcc,locations),
    type(CC,locations),
    ( isSubloc(CC,Pcc) ;          /* from wm.pl */
      isSubloc(Pcc,CC) ).        /* from wm.pl */

```

In a few special cases the Central Concept is determined independently of this algorithm. For instance, at the beginning of the dialogue, the system waits for the user to introduce the topic in her first contribution. We stipulate that the CC of the first system contribution (*Welcome to EYP. Please state your query.*) is ‘greet’. Also, if the goal is of certain particular types, CC is determined by separate predicates. Below are listed some CC filtering constraints which are not captured by the main algorithm:

- The remark about irrelevance has the same CC as the main goal attitude:

```

ccFilter(_,know(u,notrelevant(_))).

```

- The remark about topic has the topic as CC:

```

ccFilter(N,know(u,talkingAbout(CC))) :- assertz(topic(N,CC)).

```

- The remark about inconsistency has the inconsistent concept as CC:

```

ccFilter(N,know(u,inc(X,_,_))) :- assertz(topic(N,X)).

```

- The query about showing all companies has the company class as CC:

```
ccFilter(N, know(s, showall(_, Class))) :-
    ( dr(Class)
      ;
      assert(explicit(Class)) ),
    assertz(topic(N, Class)).
```

- When YP-information is supplied, the company class is CC:

```
ccFilter(N, know(u, supplyInfo(Spec))) :-
    Spec = [Class|_],
    assertz(topic(N, Class)).
```

- If the cardinality is nil, the company class is CC:

```
ccFilter(N, know(u, [cardinality(_, nil)|Spec])) :-
    Spec = [Class|_],
    assertz(topic(N, Class)),
    assert(explicit(Class)).
```

- Otherwise cardinality has the same CC as main goal attitude:

```
ccFilter(_, know(u, [cardinality(_, _)|_])).
```

- Listing all companies has the company class as CC:

```
ccFilter(N, know(u, [listall(_, _)|Spec])) :-
    Spec = [Class|_],
    assertz(topic(N, Class)).
```

- Listing the company types has the company class as CC:

```
ccFilter(N, know(u, [whattypes(_, _)|Spec])) :-
    Spec = [Class|_],
    assertz(topic(N, Class)).
```

- Giving the YP-information has the same CC as the main goal attitude:

```
ccFilter(_, know(u, ypinfo(_, _))).
```

- The query about wanting to know something else has the previous CC as CC:

```
ccFilter(N, know(s, somethingElse)) :-
    prevTopic(N, Pcc),
    assertz(topic(N, Pcc)).
```

- If the information is already given, CC is the previous CC:

```
ccFilter(N, know(u, alreadyGiven)) :-
    prevTopic(N, Pcc),
    assertz(topic(N, Pcc)).
```

- When reverting to a previous goal, CC is the topic of this goal and is explicit:

```
ccFilter(N, know(u, backto(CC))) :-
    assertz(topic(N, CC)),
    assert(explicit(CC)).
```

- When clarifying NewInfo, CC shifts to the unclear NewInfo:

```
ccFilter(N, clarify(_)) :-
    M is N - 1,
    new(M, nil(New)),
    assertz(topic(N, New)).
```

We assume that communicative intentions that belong to the same goal are thematically related to each other. Since the CC is determined separately for each communicative intention, we avoid multiple CCs by letting the main intention determine the CC and passing the other intentions as such through `ccFilter/2`.

Building the propositional content

The building of propositional content starts from NewInfo: this is realised as an elliptical contribution unless the dialogue context is such that the communicative goal cannot be conveyed to the user appropriately. In this case, NewInfo is accompanied by contextual information so that the relevant information is explicitly expressed in the contribution or unambiguously inferable from the context. Propositional content is expressed in a quasi-logical form which represents the system response in a surface oriented way: not only are the semantic predicates chosen but the representation is augmented with auxiliary predicates as well.

The main predicate for the building of the semantic representation is `propContent/3` which is as follows:

```

propContent(N,Goal,SemRepr) :-
    mood(Goal,Mood),
    getGoalConcepts(Goal,GoalConc),      /* from topic.pl      */
    initAgenda(N,GoalConc,Agenda),!,
    nl,write('Initial Agenda: '),write(Agenda),nl,nl,
    relevant(N,Goal,GoalConc,Agenda,Elliptical,RelConc),!,
    nl,write('----- Ellipsis determined '),nl,nl,
    write('Elliptical: '),write(Elliptical),nl,nl,
    nl,write('Final Agenda: '),write(RelConc),nl,
    nl,write('is Accurate, Valid, Consistent, and '),
    write('Free from false implicatures'),nl,nl,
    realise(N,Goal,RelConc,Mood,SemRepr).

```

It has a simple control structure which first determines the grammatical mood of the contribution, then initialises Agenda and checks its relevance, and finally realises the final agenda in qlf-predicates. Mood refers to the grammatical mood of the sentence. In order not to complicate the planning unnecessarily, the system assumes the prototypical relation between communicative acts and grammatical moods: interrogative mood realises the speaker's intention to know something and declarative mood the speaker's intention to let the hearer know something.

The Agenda contains the concepts that must be explicitly realised in the surface contribution. Initialisation of the Agenda consists of pushing NewInfo into Agenda, and also CC, if this is marked as explicit. If the content of the goal is not a list of concepts but a single term such as somethingElse, backto(CC), talkingAbout(CC), inc(X,0,C), showall(_,Class), ypinfo(In,Info) or supplyInfo([Class|_]), the Agenda is flagged final and initialised by the corresponding term.

```

initAgenda(N,GoalConc,Agenda) :-
    islist(GoalConc),
    news(N,News),                      /* from topic.pl      */
    explicitConcepts(ExplConc),
    append(News,ExplConc,Agenda).
initAgenda(_,GoalConc,[final,GoalConc]).

```

The predicate relevant/6 deals with the four relevance criteria. accurate/4 requires that the goal is accurately presented to the user and at least NewInfo and CC are unambiguously communicated. valid/6 requires that the hearer's intentions have

been appropriately addressed, while consistent/6 deals with the internal coherence of the set of attitudes that are to be expressed and evoked by the utterance. Finally, ffi/6 states that the contribution must be free from false implicatures: it must not lead the hearer to draw inferences that are not true. Each modification of the Agenda results in the evaluation of all the criteria. Some implemented relevance criteria are listed below.

- If Agenda contains all goal concepts and also some extra concepts not contained in GoalConc, Agenda is accurate. (That the extra concepts do not cause false implicatures or inconsistency is checked by the other tests.)

```
accurate(GoalConc,Agenda,Agenda) :-
    contains(Agenda,GoalConc).
```

- Goal Concepts that are not known (i.e. discourse referents) are added to Agenda.

```
accurate(GoalConc,Agenda,El1,AccAgenda) :-
    set_diff(GoalConc,Agenda,Diff),
    ( Diff = [],
      AccAgenda = Agenda,
      El1 = true
    ;
      notKnown(Diff,NotKnown),
      append(Agenda,NotKnown,AccAgenda),
      El1 = false ).
```

- System's answer after a user question is valid if Agenda addresses the evocative intentions of the user question.

```
validity(N,known(u,P),Agenda,ValAgenda,Val) :-
    M is N - 1,
    explEvocAtt(M,EcAtts),
    ( /* expected(known(u,P),EcAtts),!,*/
      addressEvoc(EcAtts,P,Agenda,ValAgenda,Val)
    ;
      nl,write('Agenda '),write(Agenda),nl,
      write(' and goal '),write(known(u,P)),nl,
      write(' do not match') ).
```

```
validity(_,G,Agenda,ValAgenda,Val) :-
    ( G = known(s,P)
    ;
      G = do(u,P) ),
    set_diff(P,Agenda,Diff),
```

```

( Diff = [],
  ValAgenda = Agenda,
  Val = true,!
;
  notKnown(Diff,NotKnown),
  append(Agenda,NotKnown,ValAgenda),
  Val = false,! ).

```

- System's response is free from false implicatures if Agenda conveys implicit evocative intentions of the user question.

```

isFFI(N, know(u,P), Agenda, FFIagenda, FFI) :-
  M is N - 1,
  implEvocAtt(M, EcAtts),
  expected(know(u,P), EcAtts),!,
  containImplEvoc(EcAtts,P, Agenda, FFIagenda, FFI).

```

Translation into QLF-predicates

The shallow NL front-end requires that the propositional content is syntactically and semantically fully specified when it enters the surface generator. This design criterion is in accordance with the argumentation in PLUS where it was not considered desirable that the generator should produce a set of synonymous sentences of which one is arbitrarily chosen as output. The conceptual representation of the system response is thus translated into a quasi-logical form which is close to the surface structure of English. Translation is done in two stages by the predicate `realise`: first the explicit concepts are translated into QLF-predicates via the bidirectional conceptual lexicon (`concToQLF`), and then the representation is annotated with necessary auxiliary predicates such as agreement features, pronouns etc. to fit into the requirements of the surface generator (`augment`). Annotation is a reverse operation of what was done in interpretation by ignoring some semantic predicates for the reasoning process.

```

realise(N, Goal, RelConc, Mood, SemRepr) :-
  concToQLF(RelConc, Message),!,
  append(Mood, Message, SemRepr1),
  augment(N, Goal, SemRepr1, SemRepr).

```

Following the work in PLUS, we also assume that the mapping from concepts to QLF-predicates provides important information about how the concept can be realised. Since

the dialogue manager has no access to linguistic resources, constraints on linguistic realisation must be checked indirectly. Besides mapping concepts to QLF-predicates directly via the conceptual lexicon, the predicates `concToQLF` and `augment` also provide more complex links between the propositional content and QLF-representation. For instance, they can decide on the active or passive forms of the sentence, or on which translation is preferred if a concept happens to have ambiguous translations (cf. lexical lookup in PLUS, Bego et al. (1993)). Some clauses of the predicate `concToQLF` are listed below.

```
concToQLF([ypinfo(_,Info)|_],[ok,Info]).
concToQLF([somethingElse],[ok|Pred]) :-
    concToQLF([somethingElse],Pred).
concToQLF([showall,CC],Qlf) :-
    CC =.. [_,In|_],
    concToQLF([showall(In)], [P1]),
    concToQLF([CC],P2),
    append(P1,P2,Qlf). /* [P2] */
concToQLF([Conc|Concs],[Pred|Preds]) :-
    cl(Pred,Conc), /* from cl.pl */
    concToQLF(Concs,Preds).
```

9.5.5 Surface generation

We use a Categorical Grammar based generator Lager and Black (1994) for surface generation. As already mentioned, management goals are directly generated as simple messages without reasoning about the appropriate content. Below is the result of the predicate `surfaceGen/2` for such goals, as well as the clause calling for the actual generator.

```
surfaceGen([[quit]],[['I',want,to,'quit.']].
surfaceGen([[cannot_parse]],
    [['Sorry',cannot,parse,the,'input.',
      'Could',you,please,'rephrase.']] ).
surfaceGen([[not_understand]],
    [['Sorry',cannot,'understand.', ' What',do,you,'mean?']] ).
surfaceGen([[incorrect]],
    [['There',is,something,wrong,in,your,'response.',
      'Could',you,please,'repeat?']] ).
```

```

surfaceGen([[notrelevant]],
           [['Sorry. I',do,not,understand,how,this,is,relevant,'here.']] ).

surfaceGen([SemRepr|Srs],[String|Strs]) :-
    generate(_:SemRepr,String), /* from cky_ccg_generator.p */
    surfaceGen(Srs,Strs).

```

When the surface string has been formed and given as a system response, the system returns the turn back to the user and waits for another input. Depending on the dialogue state, the system expects to get either an answer or another question, and once the user has typed in her contribution, the same cycle: accept-interpret-react starts again.

9.6 Conclusion

In this chapter we presented a protypte implementation of the principles of constructive dialogue management. The system exhibits robust features in that it is capable of dealing with the following tasks:

1. determining an appropriate continuation by reasoning about the joint purpose,
2. providing compensation if the database search results in nil,
3. specifying the goal with respect to communicative principles,
4. deciding on an appropriate Central Concept in the context,
5. planning a full sentence or ellipsis on the basis of context,
6. deciding whether a discourse referent is referred to as a definite or an indefinite NP, as a pronoun *it* or as a pronoun *one*.

A sample run of the system for the dialogue presented at the beginning of the thesis is given in Appendix 1.

Chapter 10

Discussion

10.1 Conclusions

This thesis presents a new way to deal with dialogue management in human-computer interaction. The aim is to improve the robustness of dialogue management by an approach which requires adherence to communicative principles. The approach is called ‘Constructive Dialogue Management’ and it considers dialogues as cooperative joint activities whereby the participants communicate wants and carry out plans by negotiating about the joint purpose.

Based on the empirical studies of Wizard-of-Oz dialogues carried out in the PLUS project, we formulated four requirements for robust dialogue management, and concentrated especially on the one which requires the system’s communicative competence to clarify vagueness and misunderstandings. The notion of ‘robustness’ in dialogue management is tied to the basic requirements of communication in general. We dealt with information-seeking dialogues in an application domain based on selected scenarios of finding information from the Yellow Pages, but it is claimed that the communicative principles hold for other types of dialogues as well.

The theoretical foundation for our research is Communicative Activity Analysis, a pragmatic approach to communication developed by Allwood (1976) in his dissertation and

later papers. CAA is a philosophically oriented linguistic theory, and it provides a vocabulary of concepts rather than laws or rules of conduct. Instead of formalising the whole theory of CAA, the aim in this thesis was to study the basis for successful implementation of robust dialogue management. CAA gives us the basis for identifying the type of pragmatic knowledge needed, and how this knowledge affects communication. Our contention is that the unified theory of communication provides a basis for robust dialogue management. The prototype presented in the thesis is a step towards practical validation of this claim.

CAA regards communication as cooperative activity between rational agents. In Constructive Dialogue Management we have formalised the basic concepts of CAA such as the four step analysis of communication (contact, perception, understanding and attitudinal reaction), requirements of rational ideal cooperation (joint purpose, consideration and trust), expressive and evocative dimensions of a contribution and the obligation of responsiveness, and demonstrated their importance in flexible and cooperative system architecture. Although the CAA ideas were also discussed in the PLUS project, the thesis is the first extensive implementation of the CAA principles in a dialogue manager.

We have taken a radical approach to dialogue management in two respects: we have abandoned dialogue grammar as a feasible way to deal with dialogues, and we have also abandoned speech acts and their recognition. The first decision means that structuring of dialogues is not done according to a set of structuring rules but the structure emerges as the dialogue goes on. The notion of ‘adjacency pairs’ is acknowledged as a default reaction and encoded in the expectations of an appropriate continuation, but it is not used as a constraint on the reaction. The second decision is in accordance with the current view of speech act theory: speech acts are understood as convenient abbreviations for a set of attitudes held by speakers, but it is not considered reasonable to classify them; rather the speaker’s goals and beliefs are represented as a bundle of attitudes which can be reasoned about the basis of dialogue context.

The central claim is that **dialogues are managed locally by reacting to the changed dialogue context**. The speakers’ contributions function as context up-

dates, and the appropriate reaction is formulated as a result of the evaluation of the partner's, as well as one's own goals, beliefs, and intentions with respect to the changing context. The context contains knowledge about the current dialogue situation (Central Concept and NewInfo, initiatives taken by the speakers, the speakers' attitudes), the previous contributions, world model, the specific task that is to be performed (application model), and roles and expectations of the speakers. The attitudes represent the communicative intentions of the speaker and they are of three kinds: expressive, evocative and evoked, corresponding to the attitudes that the speaker intends to express by her contribution, that she wants to evoke in the hearer and that she evokes in the hearer (the latter comprises also the presupposed attitudes which are assumed to be true if the communicative act performed by the contribution is successful). The reasoning is governed by communicative knowledge about ideal cooperation and rational activity.

The system response is based on the requirement that the evocative dimension of the user contribution and the evoked system response must match. Planning takes the form of gradual specification of the system goal on three levels: determining the relevant information to be communicated to the user (joint purpose), selecting the view-point from which this information can be presented so that its relevance in the dialogue context is clear (determining Central Concept), and determining the way of presenting the information so that the new information is conveyed but the hearer cannot draw false implicatures (selecting relevant semantic predicates for surface level realisation). Cooperativeness is implicit in the inferences that the system draws in order to fulfil the *obligation of responsiveness* and to maintain consistency of the context.

The process of finding a joint purpose takes care of the system goal formulation. The research has identified four contextual factors which are important in this: which dialogue participant has the initiative, what is the Central Concept of the dialogue, are the expectations of the previous contribution fulfilled, and has the system own goals to pursue. The system is not only a benevolent information provider, but is also able to resolve some conflicts which occur if the user contribution and the context model do not match. Conflicts can occur if the user has replied with an unrelated response in a situation where the system has had the initiative and own goals to pursue. In these

cases, the system can be persistent and repeat the goal, but at the same time, adhere to communicative principles which ensure that the user has the ultimate control of the dialogue and its topic.

The view-point from which new information in the dialogue is presented is modelled with the help of Central Concept which is an instantiated discourse referent. A distinction is drawn between aboutness and newness, the former referring to the partition of the Context Model into the Central Concept (the topic, which is currently talked about) and other known background information, and the latter referring to the new-old distinction between what is new in the dialogue context (NewInfo) and what is already known in it.

The view-point is determined so that the shift from the previous Central Concept to the intended one is thematically related: the view-points cannot shift arbitrarily but must obey coherence constraints which, for their part, are signs of the speakers' communicative competence. Thematically related shifts are formulated on the basis of domain knowledge, and encoded in nine rules. If an unrelated shift is attempted, it must be appropriately marked on the surface by a topic shift marker. Coherence of the dialogues is guaranteed locally, and global coherence is an effect that arises from the partner's evaluation of her own goals in the updated context: it is a sign of the speaker's communicative competence.

The planning of how to realise the system goal presents a new approach. The planning starts from NewInfo, and thus an elliptical contribution is the default output. A full sentence is generated only if the elliptical one does not fulfill the relevance criteria (does not communicate the goal appropriately or allows the hearer to draw unwanted implicatures).

The amount of information to be included in the surface form and the way in which it will be presented to the user are determined on the basis of CAA principles. The thesis thus bridges the generation gap by providing a uniform basis for planning. Problems associated with the distinction between strategic and tactical generation are thus avoided: the communicative principles affect the planning on all levels.

10.2 What theoretical ideas give the system its power

Our research uses insights from the huge body of research that exists on dialogue management and natural language planning and generation. However, Constructive Dialogue Management puts the whole natural language processing task in a wider context, providing a new and uniform way to treat the various phenomena that have been separately studied in previous research: it adheres to the general communicative principles that underlie the individual surface level phenomena. The uniqueness of our work arises from the way in which dialogue phenomena are handled in the light of Communicative Activity Analysis in a constantly changing dialogue context: the research has shown that a unified basis for dialogue management can be provided.

We advocate a pragmatic view-point to improve the robustness of simple service systems. Pragmatics is understood as related to communication itself, which requires dealing with the impact of contextual information on natural language processing. CAA gives us the basis for identifying the type of pragmatic knowledge needed, and how this knowledge affects communication. Especially, the four constraints of rational, cooperative communication provide us with the basic framework to deal with contributions: the communicators have a joint purpose, they obey communicative obligations by showing cognitive and ethical consideration in their contributions, and finally, they trust that the partner behaves so that the three other constraints are fulfilled. Constructive Dialogue Management emphasises the system's communicative competence, and helps the designers of dialogue systems to base their design decisions on a solid theoretical basis.

The thesis also sketches a framework in which system responses are planned based directly on the pragmatic principles, and coherence and cooperativeness of the responses are attended to. The planning overcomes the problems associated with the distinction of 'what to say' and 'how to say' by gradual specification of the system goal and emphasising the new information that is to be presented to the user. The default contribution is an elliptical contribution, and the full sentence is generated only if the contextual information is needed to convey the goal appropriately or block false implicatures.

10.3 Future Directions

Being the first attempt to develop comprehensive computational dialogue management in the CAA framework, the work can be extended in different directions. Four open questions are of the most interest:

- How can the formalisation of CAA be extended to incorporate other types of dialogues (e.g. task-oriented dialogues, advisory dialogues) and other aspects of communication (e.g. reasoning about the system's role, the task it is involved in, and the expertise level of the user). In the present system these are 'hardcoded' in the design of a cooperative servant which provides information from the Yellow Pages.
- What kind of effects do knowledge representation and different reasoning methods have on the system's efficiency? The use of abductive reasoning in determining the joint purpose and the Central Concept is under investigation, and faster algorithms should be studied.
- How can the CAA principles be applied to conflict situations?
- How can the CAA principles be applied to multi-partner dialogues and spoken dialogues, where the basic communicative requirements contact and perception play a bigger role?

Bibliography

- Ahrenberg, L., Jönsson, A., & Thureé, Å. (1994). Customizing interaction for natural language interfaces. In *Proceedings of the Workshop on Pragmatics in Dialogue Management*, pp. 21–38. XIV Scandinavian Conference of Linguistics.
- Allen, J. F. (1983). Recognizing intentions from natural language utterances. In Brady, M. & Berwick, B. (Eds.), *Computational Models of Discourse*. MIT Press, Cambridge, Mass.
- Allen, J. F. & Perrault, C. R. (1980). Analyzing intention in utterances. *Artificial Intelligence*, 15, 143–178.
- Allen, S. (1990). The Wizard's Apprentice. Part 1: A program to help analyse natural language dialogues. Department of Computer Science, University of Liverpool. Undergraduate project supervised by D. Diaper.
- Allwood, J. (1976). *Linguistic Communication as Action and Cooperation*. Department of Linguistics, University of Gothenburg. Gothenburg Monographs in Linguistics 2.
- Allwood, J. (1978). On the analysis of communicative action. In Brenner, M. (Ed.), *The Structure of Action*. Basil Blackwell, Oxford.
- Allwood, J. (1984). On relevance in spoken interaction. In Kjellmer, G. & Bäckman, S. (Eds.), *Papers on Language and Literature. Gothenburg Studies in English 60*. Department of English, University of Gothenburg.
- Allwood, J. (1992). On dialogue cohesion. Gothenburg Papers in Theoretical Linguistics 65, University of Gothenburg.

- Allwood, J. (1993). Feedback and language acquisition. Gothenburg Papers in Theoretical Linguistics 68, University of Gothenburg.
- Allwood, J. & Bunt, H. C. (1992). Pragmatics in PLUS. Theoretical Underpinnings. PLUS Project Report.
- Allwood, J., Dahlgren, H., Haglund, B., Lager, T., & Nivre, J. (1991). On robustness. PLUS working paper.
- Allwood, J. & Haglund, B. (1991). Communicative activity analysis of a wizard of oz experiment. Tech. rep. PLUS working paper, Department of Linguistics, University of Gothenburg.
- Allwood, J., Nivre, J., & Ahlsén, E. (1992). On the semantics and pragmatics of linguistic feedback. *Journal of Semantics*, 9, 1–29.
- Appelt, D. (1985). *Planning Natural Language Utterances*. Cambridge University Press, Cambridge. Studies in Natural Language Processing.
- Austin, J. L. (1962). *How To Do Things With Words*. Clarendon Press, Oxford.
- Beaugrande, de, R. & Dressler, W. (1981). *Introduction to Text Linguistics*. Longman, London.
- Bego, H., Beskow, B., Bunt, H., Derain, A., Horel, L., Jokinen, K., Kraaij, W., Pernel, D., & Tabuteau, G. (1992). Pragmatic Knowledge in PLUS. Part III - Pragmatic Rules. PLUS deliverable.
- Bego, H., Bilange, E., Black, W., Bunt, H. C., Dols, F., Jokinen, K., Kraaij, W., Underwood, N., Neal, P., Herrero, J., Hatala, M., & Cunningham, H. (1993). Demonstrations of the first prototypes. Tech. rep. D3.1, PLUS deliverable.
- Beun, R.-J. (1989). *The Recognition of Declarative Questions in Information Dialogues*. Katholieke Universiteit Brabant.
- Beun, R.-J. & Bunt, H. C. (1987). Investigating linguistic behaviour in information dialogues with a computer. Tech. rep. 22, IPO Annual Progress Report.

- Bilange, E. (1991). A task independent oral dialogue model. In *Proceedings of the European ACL*. Italy.
- Bilange, E. (1992). *Dialogue personne-machine. Modélisation et réalisation informatique*. Hermès, Paris.
- Black, W. J., Bunt, H. C., Dols, F., Donzella, C., Ferrari, G., Haidan, R., Imlah, W. G., Jokinen, K., Lager, T., Lancel, J.-M., Nivre, J., Sabah, G., & Wachtel, T. (1991). A pragmatics-based language understanding system. Esprit Conference paper.
- Black, W. J. & Jokinen, K. (1991). Robust understanding of user turns in the corpus. PLUS working paper.
- Black, W. & Cunningham, H. (1992). The plus surface generator design. PLUS Project Report.
- Brachman, R. J., Levesque, H. J., & Reiter, R. (Eds.). (1992). *Knowledge Representation*. A Bradford Book, The MIT Press, Cambridge, Massachusetts.
- Brown, P. & Levinson, S. (1978). Universals in language use: politeness phenomena. In Goody, E. (Ed.), *Questions and Politeness: Strategies in Social Interaction*, pp. 56–311. Cambridge University Press.
- Bunt, H. C. (1990). DIT - dynamic interpretation in text and dialogue. In Kálmán, L. & Pólos, L. (Eds.), *Papers from the Second Symposium on Language and Logic*. Akademiai Kiadó, Budapest.
- Bunt, H. C. (1991). Dynamic interpretation and dialogue theory. In *The Second Venaco Workshop on Multimodal Dialogue*. Acquafredda di Maratea.
- Bunt, H. C., Beun, R. J., Dols, F. J. H., van der Linden, J. A., & thoe Schwartzenberg, G. O. (1984). The TENDUM dialogue system and its theoretical basis. Tech. rep. 19, IPO Annual Report.
- Burton, A. & Steward, A. P. (1993). Effects of linguistic sophistication on the usability of a natural language interface. *Interacting with Computers*, 5:1, 31–59.

- Carletta, J. (1991). A state-based representation for belief modelling in dialogue. In *Workshop on Discourse Coherence*. University of Edinburgh.
- Carlson, L. (1983). *Dialogue Games*. Reidel, Dordrecht.
- Carlson, L. (1988). Questions of identity in discourse. In Meyer, M. (Ed.), *Questions and Questioning*, pp. 144–181. Walter de Gruyter, Berlin.
- Cavalli, V., Dahlgren, H., Donzella, C. E., Fujol, S., & Godin, C. (1992a). System evaluation criteria and test environment. Tech. rep. D2.2, PLUS deliverable.
- Cavalli, V., Guessoum, A., Lager, T., Meyer, R., Nivre, J., & Underwood, N. (1992b). Pragmatic Knowledge in PLUS. Part I - World and Application Model. PLUS deliverable.
- Cawsey, A. (1990). Generating explanatory discourse. In Dale, R., Mellish, C., & Zock, M. (Eds.), *Current research in Natural Language Generation*, pp. 75–101. Academic Press, London.
- Cawsey, A. (1993). *Explanation and Interaction. The Computer Generation of Explanatory Dialogues*. A Bradford Book, The MIT Press, Cambridge, Massachusetts.
- Chafe, W. (1976). Givenness, contrastiveness, definiteness, subjects, and topics. In Li, C. (Ed.), *Subject and Topic*. Academic Press.
- Clark, H. H. & Haviland, S. E. (1977). Comprehension and the given-new contract. In Freedle, R. O. (Ed.), *Discourse Production and Comprehension, Vol.1*. Ablex.
- Clark, H. H. & Wilkes-Gibbs, D. (1990). Referring as a collaborative process. In *Intentions In Communication*. Cambridge, Mass: MIT Press.
- Cohen, P. R. & Levesque, H. J. (1985). Speech acts and rationality. In *Proceedings of 23rd Annual Meeting of the Association for Computational Linguistics*, pp. 49–59.
- Cohen, P. R. & Levesque, H. J. (1990a). Persistence, intention, and commitment. In Cohen, P. R., Morgan, J., & Pollack, M. E. (Eds.), *Intentions in Communication*, pp. 33–69. A Bradford Book. The MIT Press. Cambridge, Massachusetts.

- Cohen, P. R. & Levesque, H. J. (1990b). Rational interaction as the basis for communication. In Cohen, P. R., Morgan, J., & Pollack, M. E. (Eds.), *Intentions in Communication*, pp. 221–255. A Bradford Book. The MIT Press. Cambridge, Massachusetts.
- Cohen, P. R. & Perrault, C. R. (1979). Elements of a plan-based theory of speech acts. *Cognitive Science*, 3, 177–212.
- Dahlbäck, N. & Jönsson, A. (1989). Empirical studies of discourse representations for natural language interfaces. Tech. rep. LiTH-IDA-R-89-33, Department of Computer Science and Information Science, University of Linköping. Also in Proceedings of the 4th EACL Conference, Manchester, April 1989.
- Dahlbäck, N., Jönsson, A., & Ahrenberg, L. (1993). Wizard of Oz Studies - Why and How. In *Proceedings from the 1993 International Workshop on Intelligent User Interfaces*, pp. 951–957. Orlando, Florida.
- Dale, R. (1988). The generation of subsequent referring expressions in structured discourses. In Zock, M. & Sabah, G. (Eds.), *Advances in Natural Language Generation: An Interdisciplinary Perspective, Vol 2*, pp. 58–75. Pinter Publishers Ltd.
- Dale, R. & Haddock, N. (1991). Content determination in the generation of referring expressions. *Computational Intelligence*, 7, 252–265.
- Dale, R., Mellish, C., & Zock, M. (1990). Introduction. In Dale, R., Mellish, C., & Zock, M. (Eds.), *Current Research in Natural Language Generation*, pp. 1–16. Academic Press, London.
- Danlos, L. (1987). *The Linguistic Basis of Text Generation*. Cambridge University Press.
- Derr, M. A. & McKeown, K. R. (1984). Using focus to generate complex and simple sentences. In *Proceedings of the 10th COLING*, pp. 319–326.

- Diaper, D. (1986). Identifying the knowledge requirements of an expert system's natural language processing interface. In *People and Computers: Designing for Usability*, pp. 263–280. Cambridge University Press.
- Diaper, D. (1989). Giving HCI away. In *People and Computers V*, pp. 109–117. Cambridge University Press.
- Fraser, N. M. & Gilbert, G. N. (1991). Simulating speech systems. *Computer Speech and Language*, 5, 81–99.
- Gallagher, J. & Guessoum, A. (1992). Abduction and integrity recovery for the control of knowledge base interaction. In *Pre-prints of the ESPRIT/PLUS Workshop on Computational Pragmatics*. Alghero, Sardinia.
- Galliers, J. R. (1989). A theoretical framework for computer models of cooperative dialogue, acknowledging multi-agent conflict. Tech. rep. 172, University of Cambridge, Computer Laboratory.
- Garrod, S. C. & Sanford, A. (1983). Topic dependent effects in language processing. In d'Arcais, G. B. F. & Jarvella, R. J. (Eds.), *The Process of Language Understanding*, pp. 271–296. John Wiley and Sons Ltd.
- Gazdar, G. (1979). *Pragmatics: Implicature, Presupposition and Logical Form*. Academic Press, New York.
- Gazdar, G. & Mellish, C. (1989). *Natural Language Processing in PROLOG*. Addison-Wesley.
- Goffman, E. (1970). *Strategic Interaction*. Blackwell, Oxford.
- Goffman, E. (1976). Replies and responses. *Language in Society*, 5, 257–313.
- Grice, P. H. (1957). Meaning. In Strawson, P. F. (Ed.), *Philosophical Logic*. Oxford University Press.
- Grice, P. H. (1975). Logic and conversation. In Cole, P. & Morgan, J. L. (Eds.), *Syntax and Semantics Vol. 3: Speech Acts*. Academic Press.

- Grishman, R. & Kittredge, R. (Eds.). (1986). *Analyzing Language in Restricted Domains: Sublanguage Description and Processing*. Lawrence Erlbaum, Hillsdale, New Jersey.
- Grosz, B. J. (1977). The representation and use of focus in dialogue understanding. Tech. rep. 151, SRI International, Menlo Park.
- Grosz, B. J., Joshi, A. K., & Weinstein, S. (1983). Providing a unified account of definite noun phrases in discourse. In *Proceedings of 21st Annual Meeting of Association of Computational Linguistics*, pp. 44 – 50.
- Grosz, B. J. & Sidner, C. L. (1986). Attention, intentions, and the structure of discourse. *Computational Linguistics*, 12(3), 175–204.
- Guessoum, A. & Lloyd, J. W. (1990). Updating knowledge bases. *New Generation Computing*, 8(1), 71–89.
- Guessoum, A. & Lloyd, J. W. (1991). Updating Knowledge Bases II. *New Generation Computing*, 10(1), 73–100.
- Halliday, M. A. K. (1967). Notes on transitivity and theme in English. *Journal of Linguistics*, 3(2), 199–244.
- Halliday, M. A. K. (1973). *Explorations in the Functions of Language*. Arnold, London.
- Halliday, M. A. K. & Hasan, R. (1976). *Cohesion in English*. Longman, London.
- Hayes, P. J. & Reddy, D. R. (1983). Steps toward graceful interaction in spoken and written man-machine communication. *International Journal of Man-Machine Studies*, 19, 231–284.
- Hobbs, J. R. (1979). Coherence and coreference. *Cognitive Science*, 3, 67–90.
- Hobbs, J. R., Stickel, M., Appelt, D., & Martin, P. (1990). Interpretation as abduction. Technical Note No. 499, Artificial Intelligence Center, SRI International, Menlo Park.

- Horacek, H. (1992). An integrated view of text planning. In Dale, R., Hovy, E., Rösner, D., & Stock, O. (Eds.), *Aspects of Automated Natural Language Generation*, pp. 29–44. Springer-Verlag, Berlin. Proceedings of the 6th International Workshop on Natural Language Generation, Trento, Italy.
- Hovy, E. H. (1986). Some pragmatic decision criteria in generation. In Kempen, G. (Ed.), *Natural Language Generation: Recent Results in Artificial Intelligence, Psychology, and Linguistics*, pp. 3–18. Kluwer Academic Publishers, Boston, MA.
- Hovy, E. H. (1988a). *Generating Natural Language Under Pragmatic Constraints*. Lawrence Erlbaum Associates, Hillsdale, New Jersey.
- Hovy, E. H. (1988b). Planning coherent multisentential text. In *26th ACL Conference Proceedings*, pp. 163–169. Buffalo.
- Hovy, E. H. (1988c). Two types of planning in language generation. In *26th ACL Conference Proceedings*, pp. 179–186. Buffalo.
- Hovy, E. H. (1990). Unresolved issues in paragraph planning. In Dale, R., Mellish, C., & Zock, M. (Eds.), *Current Research in Natural Language Generation*, pp. 17–47. Academic Press, London.
- Hovy, E. H. (1991). Approaches to the planning of coherent text. In Paris, C. L., Swartout, W. R., & Moore, W. C. (Eds.), *Natural Language Generation in Artificial Intelligence and Computational Linguistics*, pp. 83–102. Kluwer Academic Publishers, Norwell, MA.
- Hovy, E. H. & McCoy, K. F. (1989). Focusing your RST: A step toward generating coherent multisentential text. In *Proceedings of the 11th Cognitive Science Conference*, pp. 667–674. Ann Arbor, August 1989. Also available as USC/Information Sciences Institute Research Report RS-89-246.
- Jokinen, K. (1991). Preliminary analysis of WOZ-dialogues. PLUS working paper.
- Jokinen, K. (1993a). Categorical Grammar for Cobalt. Tech. rep. Cobalt project Deliverable 5.3 (Grammar and Lexicon Enhancement) Part 1, UMIST, Manchester.

- Jokinen, K. (1993b). Reasoning about coherent and cooperative system responses. In *The Fourth European Workshop on Natural Language Generation Preprints*, pp. 115–126. Pisa, April 1993.
- Jokinen, K., Jones, D., & Black, W. J. (1991). Dialogue Grammar in PLUS. PLUS working paper.
- Jokinen, K., Kraaij, W., Meyer, R., Nivre, J., Pernel, D., Bunt, H., & Godin, C. (1992). Pragmatic Knowledge in PLUS. Part II - Discourse Model. PLUS deliverable.
- Joshi, A., Webber, B. L., & Weischedel, R. M. (1984). Preventing false inferences. In *Proceedings of the 10th COLING*, pp. 134–138.
- Joshi, A. K. (1982). Mutual beliefs in question answering systems. In Smith, N. (Ed.), *Mutual Belief*, pp. 181–221. Academic Press, New York.
- Kamp, H. & Reyle, U. (1993). *From discourse to logic: Introduction to model theoretic semantics of natural language, formal logic and discourse representation theory*. Kluwer Academic Publishers, Dordrecht.
- Kantrowitz, M. & Bates, J. (1992). Integrated natural language generation systems. In Dale, R., Hovy, E., Rösner, D., & Stock, O. (Eds.), *Aspects of Automated Natural Language Generation*, pp. 13–28. Springer-Verlag, Berlin. Proceedings of the 6th International Workshop on Natural Language Generation, Trento, Italy.
- Kaplan, S. J. (1983). Co-operative response from a portable natural language system. In Brady, M. & Berwick, B. (Eds.), *Computational Models of Discourse*, pp. 167–208. MIT Press.
- Kittredge, R., Korelsky, T., & Rambow, O. (1991). On the need for domain communication knowledge. *Computational Intelligence*, 7, 305–314.
- Kuppevelt, van, J. (1991). *Topic en Comment. Expliciete en Impliciete Vraagstelling in Discourse*. Ph.D. thesis, University of Nijmegen.

- Lager, T. & Black, W. J. (1994). Bidirectional Incremental Generation and Analysis with Categorical Grammar and Indexed Quasi-Logical Form. A paper presented in the 7th International Workshop on Natural Language Generation.
- Levinson, S. C. (1983). *Pragmatics*. Cambridge University Press, Cambridge.
- Litman, D. (1985). Plan recognition and discourse analysis: an integrated approach for understanding dialogues. Tech. rep. 170, Department of Computer Science, University of Rochester.
- Litman, D. & Allen, J. F. (1987). A plan recognition model for subdialogues. *Cognitive Science*, 11, 163–200.
- Litman, D. J. & Allen, J. F. (1984). A plan recognition model for clarification subdialogues. In *Proceedings of the 10th COLING*, pp. 302–311.
- Loo, van, W. & Bego, H. (1994). Agent tasks and dialogue management. In Jokinen, K. (Ed.), *Proceedings of the Workshop on Pragmatics in Dialogue Management*. XIV Scandinavian Conference of Linguistics.
- Luff, P., Gilbert, N., & Frohlich, D. (Eds.). (1990). *Computers and Conversation*. Academic Press, London.
- Lyons, J. (1977). *Semantics. Vols. 1 and 2*. Cambridge University Press, Cambridge.
- Malinowski, B. (1923). The problem of meaning in primitive languages. In Ogden, C. K. & Richards, I. A. (Eds.), *The Meaning of Meaning*. Routledge and Kegan Paul, London.
- Mann, W. C. & Matthiessen, C. M. I. M. (1983). Nigel: a systemic grammar for text generation. Tech. rep. RR-83-105, USC/ISI Research Report.
- Mann, W. C. & Moore, J. (1981). Computer generation of multiparagraph text. *American Journal of Computational Linguistics*, 7, 17–29.
- Mann, W. C. & Thompson, S. A. (1987). Rhetorical structure theory: toward a functional theory of text organization. Tech. rep. RR-87-190, USC/ISI Research Report.

- McCoy, K. (1984). Correcting object-related misconceptions: how should the system respond. In *Proceedings of the 10th COLING-84, Stanford*, pp. 444–447.
- McCoy, K. & Cheng, J. (1990). Focus of attention: constraining what can be said next. In Paris, C. L., Swartout, W. R., & Moore, W. C. (Eds.), *Natural Language Generation in Artificial Intelligence and Computational Linguistics*, pp. 103–124. Kluwer Academic Publishers, Norwell, MA.
- McDonald, D. (1983). Natural language generation as a computational problem. In Brady, M. & Berwick, B. (Eds.), *Computational Models of Discourse*, pp. 209–265. MIT Press, Cambridge, Mass.
- McDonald, D. (1993). Does natural language generation start from a specification?. In Horacek, H. & Zock, M. (Eds.), *New Concepts in Natural Language Generation*, pp. 275–278. Pinter Publishers, London.
- McKeown, K. (1985). *Text Generation: Using discourse strategies and focus constraints to generate natural language text*. Cambridge University Press, Cambridge.
- McKeown, K. R. & Swartout, W. R. (1988). Language generation and explanation. In Zock, M. & Sabah, G. (Eds.), *Advances in Natural Language Generation: An Interdisciplinary Perspective, Vol 1*, pp. Vol 1, 1–51. Pinter Publishers Ltd.
- McKevitt, P., Partridge, D., & Wilks, Y. (1992). Approaches to natural language discourse processing. *Artificial Intelligence Review*, 6:4, 333–364.
- Meteer, M. (1991). Bridging the generation gap between text planning and linguistic realization. *Computational Intelligence*, 7, 296–304.
- Meyer, R. & Guessoum, A. (1994). Abduction and conceptual filtering for pragmatic reasoning in dialogue systems. In *Proceedings of the Workshop on Pragmatics in Dialogue Management*, pp. 157–174. XIV Scandinavian Conference of Linguistics.
- Moeschler, J. (1989). *Modélisation du dialogue*. Hermès, Paris.

- Mooney, D. J., Carberry, S., & McCoy, K. F. (1991). Capturing high-level structure of naturally occurring, extended explanations using bottom-up strategies. *Computational Intelligence*, 7, 334–356.
- Moore, J. D. & Paris, C. L. (1989). Planning text for advisory dialogues. In *Proceedings of ACL*, pp. 67–75. 27th Annual Meeting, Vancouver, Canada.
- Moore, J. D. & Paris, C. L. (1993). Planning text for advisory dialogues: capturing intentional and rhetorical information. *Computational Linguistics*, 19(4), 651–694.
- Moore, J. D. & Pollack, M. E. (1992). A problem for RST: The need for multi-level discourse analysis. *Computational Linguistics*, 18(4), 537–544.
- Moore, J. D. & Swartout, W. R. (1990). A reactive approach to explanation: taking the user's feedback into account. In Paris, C. L., Swartout, W. R., & Moore, W. C. (Eds.), *Natural Language Generation in Artificial Intelligence and Computational Linguistics*, pp. 3–48. Kluwer Academic Publishers, Norwell, MA.
- Morris, C. W. (1938). Foundations of the theory of signs. In Neurath, O., Carnap, R., & Morris, C. (Eds.), *International Encyclopedia of Unified Science*, pp. 77–138. University of Chicago Press, Chicago. Reprinted as C. W. Morris *Writings on the General Theory of Signs*. Mouton, The Hague, 1971.
- Murray, W. R. (1989). Control for intelligent tutoring systems: a blackboard-based dynamic instructional planner. In Bierman, D., Breuker, J., & Sandberg, J. (Eds.), *Artificial Intelligence and Education*, pp. 150–168. IOS, Amsterdam.
- Nivre, J. (Ed.)(1992a). Corpus Collection and Analysis. Tech. rep. D2.1 PLUS deliverable.
- Nivre, J. (1992b). *Situations, Meaning and Communication*. Department of Linguistics, University of Gothenburg. Gothenburg Monographs in Linguistics 11.
- Paris, C. (1988). Tailoring object descriptions to a user's level of expertise. *Computational Linguistics*, 14, 64–78.

- Perrault, C. R. (1987). An application of Default logic to Speech Act Theory. Tech. rep. CSLI-87-90, CSLI, SRI International.
- Pignataro, V. (1988). A computational approach to topic and focus in a production model. In *Proceedings of COLING Conference*, pp. 515–517.
- Pollack, M. E., Hirshberg, J., & Webber, B. L. (1982). User participation in the reasoning process of expert systems. Tech. rep. CIS-82-10, University of Pennsylvania. A short version of the report appears in the *Proceedings of the second National Conference on Artificial Intelligence*, 1982.
- Quirk, R., Greenbaum, S., Leech, G., & Svartvik, J. (1972). *A Grammar of Contemporary English*. Longman, London.
- Redeker, G. (1990). Ideational and pragmatic markers of discourse structure. *Journal of Pragmatics*, 17, 367–381.
- Reichman, R. (1985). *Getting Computers to Talk Like You and Me. Discourse Context, Focus, and Semantics (An ATN Model)*. A Bradford Book. The MIT Press. Cambridge, Massachusetts.
- Reiter, E. (1990). Generating descriptions that exploit a user’s domain knowledge. In Dale, R., Mellish, C., & Zock, M. (Eds.), *Current Issues in Natural Language Generation*, pp. 257–285. Academic Press.
- Rentier, G., van der Sloot, K., Black, W., Neil, P., & Cunningham, H. (1993). English NL Engine Demonstration. Tech. rep. D3.2, PLUS deliverable.
- Roulet, E. (1986). Complétude interactive et mouvements discursifs. *Cahiers de Linguistique Française*, 7, 189–206.
- Rubinoff, R. (1992). Integrated text planning and linguistic choice by annotating linguistic structures. In Dale, R., Hovy, E., Rösner, D., & Stock, O. (Eds.), *Aspects of Automated Natural Language Generation*, pp. 45–56. Springer-Verlag, Berlin. Proceedings of the 6th International Workshop on Natural Language Generation, Trento, Italy.

- Sacks, H., Schegloff, E. A., & Jefferson, G. (1974). A simplest systematics for the organization of turn-taking in conversation. *Language*, 50(4), 696–735.
- Scha, R. & Polanyi, L. (1988). An augmented context-free grammar for discourse. In *Proceedings of the 12th International Conference on Computational Linguistics*, pp. 22–27. COLING-88.
- Schegloff, E. A. & Sacks, H. (1973). Opening up closing. *Semiotica*, 7(4), 289–327.
- Searle, J. R. (1969). *Speech Acts*. Cambridge University Press, Cambridge.
- Sgall, P., Hajicová, E., & Benesova, E. (1973). *Topic, Focus and Generative Semantics*. Scriptor Verlag, Kronberg Taunus.
- Sibun, P. (1991). Generating text without trees. *Computational Intelligence*, 7(4).
- Sidner, C. L. (1979). Towards a computational theory of definite anaphora comprehension in english discourse. Tech. rep. 537, Artificial Intelligence Laboratory, MIT, Cambridge, Massachusetts.
- Sinclair, J. M. & Coulthard, R. M. (1975). *Towards an Analysis of Discourse: the English Used by Teachers and Pupils*. Oxford University Press, London.
- Smedt, de, K. & Kempen, G. (1987). Incremental sentence production, self-correction and co-ordination. In Kempen, G. (Ed.), *Natural Language Generation: New Results in Artificial Intelligence, Psychology and Linguistics*. Martinus Nijhoff.
- Sperber, D. & Wilson, D. (1987). *Relevance: Communication and Cognition*. Basil Blackwell, Oxford.
- Stede, M. (1993). Lexical choice criteria in language generation. In *Proceedings of the 6th EACL Conference, Utrecht*, pp. 454–459.
- Steedman, M. (1991). Structure and intonation. *Language*, 67(2), 260–296.
- Suchman, L. (1987). *Plans and Situated Actions*. Cambridge University Press, Cambridge.

- Thompson, H. (1979). Strategy and tactics: a model for language production. In *Papers from the 13th meeting*, pp. 651–668. Chicago Linguistics Society.
- Tsujii, J., Ananiadou, S., Arad, I., & Sekine, S. (1992). Linguistic knowledge acquisition from corpora. In *2nd International Workshop on Fundamental Research for the Future Generation of Natural Language Processing, FG/NLP Proceedings*, pp. 61–81. Centre for Computational Linguistics, UMIST, Manchester.
- Underwood, N. & Black, W. J. (1994). More local pragmatics problems and their resolution via abduction. In Jokinen, K. (Ed.), *Proceedings of the Workshop on Pragmatics in Dialogue Management*, pp. 175–192. XIV Scandinavian Conference of Linguistics.
- Veen, van de, E. (1994). Incremental generation for highly interactive dialogues. In Jokinen, K. (Ed.), *Proceedings of the Workshop on Pragmatics in Dialogue Management*, pp. 111–120. XIV Scandinavian Conference of Linguistics.
- Vilkuna, M. (1989). *Free Word Order in Finnish. Its Syntax and Discourse Functions*. Suomalaisen Kirjallisuuden Seura, Helsinki.
- Wachtel, T. (1986). Pragmatic sensitivity in natural language interfaces and the structure of conversation. In *Proceedings of COLING'86*, pp. 35–41. Bonn, Germany.
- Wahlster, W., Marburger, H., Jameson, A., & Buseman, S. (1983). Over-answering yes-no questions: Extended Responses in a NL Interface to a Vision System. In *Proceedings of the 8th IJCAI Conference*, pp. 643–646.
- Wallace, M. D. & Anderson, T. J. (1993). Approaches to interface design. *Interacting with Computers*, 5:3, 259–278.
- Wallis, J. W. & Shortliffe, E. H. (1984). Customised explanations using causal knowledge. In Buchanan, B. G. & Shortliffe, E. H. (Eds.), *Rule Based Expert Systems*, pp. 371–390. Addison-Wesley, Reading, Mass.
- Webber, B. L. (1987). Question answering. In Shapiro, S. C. (Ed.), *Encyclopedia of Artificial Intelligence*, pp. 814–822. Wiley.

Webber, B. L. & Joshi, A. (1982). Taking the initiative in Natural Language Data Base Interactions: Justifying Why. In *Proceedings of COLING'82*, pp. 413–418. North-Holland Publishing Company, Academia.

Weiner, J. L. (1980). BLAH, a system which explains its reasoning. *Artificial Intelligence*, 15, 19–48.

Appendix 1

A sample run.

Script started on Sun Jul 17 11:58:07 1994

21-skye> go

SICStus 2.1 #7: Mon Feb 8 15:47:57 GMT 1993

```
{consulting /usr/export/home/neist/kris/thesis/implement/caa.pl...}
{consulting /usr/export/home/neist/kris/thesis/implement/utils.pl...}
{/usr/export/home/neist/kris/thesis/implement/utils.pl consulted, 190 msec 32720 bytes}
{consulting /usr/export/home/neist/kris/thesis/implement/cl.pl...}
{/usr/export/home/neist/kris/thesis/implement/cl.pl consulted, 130 msec 13600 bytes}
{consulting /usr/export/home/neist/kris/thesis/implement/jointP.pl...}
{/usr/export/home/neist/kris/thesis/implement/jointP.pl consulted, 370 msec 37840 bytes}
{consulting /usr/export/home/neist/kris/thesis/implement/rp.pl...}
{consulting /usr/export/home/neist/kris/thesis/implement/contexts...}
{/usr/export/home/neist/kris/thesis/implement/contexts consulted, 140 msec 9232 bytes}
{/usr/export/home/neist/kris/thesis/implement/rp.pl consulted, 750 msec 71888 bytes}
{consulting /usr/export/home/neist/kris/thesis/implement/topic.pl...}
{/usr/export/home/neist/kris/thesis/implement/topic.pl consulted, 280 msec 26400 bytes}
{consulting /usr/export/home/neist/kris/thesis/implement/wm.pl...}
{/usr/export/home/neist/kris/thesis/implement/wm.pl consulted, 440 msec 77264 bytes}
{consulting /usr/export/home/neist/kris/thesis/implement/ypm.pl...}
{consulting /usr/export/home/neist/kris/thesis/implement/data.pl...}
{/usr/export/home/neist/kris/thesis/implement/data.pl consulted, 130 msec 12048 bytes}
{/usr/export/home/neist/kris/thesis/implement/ypm.pl consulted, 450 msec 39984 bytes}
{consulting /usr/export/home/neist/kris/thesis/implement/initial.pl...}
{/usr/export/home/neist/kris/thesis/implement/initial.pl consulted, 20 msec 2400 bytes}
{/usr/export/home/neist/kris/thesis/implement/caa.pl consulted, 2940 msec 326000 bytes}
| ?- st.
```

Welcome to EYP. Please state your query.

|: 'I need a car'.

Parse:

```
[i(_844),num(_844,sg),case(_844,nom),def(_844),need(_824,_844,_826),
stype(_824,finf),tense(_824,pres-time),num(_824,sg),pers(_824,st),
forsome(_826),car(_826),num(_826,sg),case(_826,acc),kind(_826,countn)]
```

Translation:

```
[user(u),needEvent(_824,_844,_826),car(_826)]
```

User Goal:

```
[know(s,[wantEvent(w,u,h),haveEvent(h,u,c),user(u),car(c)])]
```

Possible user shifts:

```
[needEvent(n,u,c)^coherent,car(c)^coherent]
```

Central Concept: needEvent(n,u,c)

NewInfo: [user(u),needEvent(n,u,c),car(c)]

Joint purpose:

```
[wantEvent(w,u,h),haveEvent(h,u,c),user(u),car(c)]
```

```

AM headings: [carGarage,carHireCompany]

Too many headings

Sys Goal:
1      know(s,[wantEvent(w,u,d),car(c),user(u),hireEvent(r,u,c,_4007),
buyEvent(b,u,c,_4014),disj(d,b,r)])

NewInfo:
[disj(d,b,r)]

----- Entering Response Planner

----- Checking system obligations

Obligated goal:
1      know(s,[wantEvent(w,u,d),car(c),user(u),hireEvent(r,u,c,_4007),
buyEvent(b,u,c,_4014),disj(d,b,r)])

Goal 1 to be realised:
know(s,[wantEvent(w,u,d),car(c),user(u),hireEvent(r,u,c,_4007),
buyEvent(b,u,c,_4014),disj(d,b,r)])

----- Determining Central Concept

Possible system shifts:
[car(c)^coherent]

Central Concept: car(c)

----- Determining Propositional Content

Initial Agenda:  [disj(d,b,r)]

accurate: [disj(_11142,b,r),buyEvent(b,u,c,_4014),hireEvent(r,u,c,_4007)]

valid: [disj(d,b,r),buyEvent(b,u,c,_4014),hireEvent(r,u,c,_4007),
wantEvent(w,u,d),car(c),user(u)]

consistent: [disj(d,b,r),buyEvent(b,u,c,_4014),hireEvent(r,u,c,_4007),
wantEvent(w,u,bd),car(c),user(u)]

----- Ellipsis determined

Elliptical: no_ellipsis

Final Agenda:
[disj(d,b,r),buyEvent(b,u,c,_4014),hireEvent(r,u,c,_4007),
wantEvent(w,u,d),car(c),user(u)]

is Accurate, Valid, Consistent, and Free from false implicatures

----- Leaving Response Planner

```


The semantic representation is:

```
interrog(_12187)
or(_12240,b,r)
buy(b,u,c)
rent(r,u,c)
want(w,u,d)
stype(w,finf)
tense(w,pres-time)
num(w,sg)
pers(w,nd)
one(c)
case(c,acc)
num(c,sg)
you(u)
```

----- Surface Generation:

Surface String:

Not yet connected to a surface generator.

|: 'Rent'.

Parse:

```
[rent(_15038,_15039,_15040),stype(_15038,finf),tense(_15038,pres-time),
num(_15038,_15021),pers(_15038,_15016)]
```

Translation:

```
[hireEvent(_15038,_15039,_15040,_15216)]
```

User Goal:

```
[know(s,[wantEvent(w,u,r),hireEvent(r,u,c,_15464),user(u),car(c)])]
```

Possible user shifts:

```
[car(c)^cont]
```

Central Concept: car(c)

NewInfo: [hireEvent(r,u,c,_15449)]

Joint purpose:

```
[wantEvent(w,u,r),hireEvent(r,u,c,_17973),user(u),car(c)]
```

AM headings: [carHireCompany]

Query: [carHireCompany,_18888,_18886,_18884,_18882,_18880]

Specs: []

Sys Goal:

```
1      know(s,[hireEvent(r,u,c,_17973),location(r,_20616)])
2      know(u,[cardinality(setOf(11,[[[24hr Airport Chauffeur
Drive,Kings House King St West,Manchester M3,061-798-7777],[ABC Car
and Van Hire,Viola Street,Bolton,0204 50075],[Avis
Rent-a-Car,International Arrival Hall,Manchester Airport M22,061-436
2020],[Boundary Garage 1972,Bury Road Breightmet,Bolton,0204
```

28607/35103],[British Car Rental,c/o Lex Mead Manchester Road,Bolton,0204 31183],[Budget Rent-a-Car,Bradford House 287-289 Manchester Road,Bolton,0204 391611],[Budget Rent-a-Car,In terminal,Manchester Airport M22,061 499 3042],[EuroDollar Rent-a-Car,Arrivals Hall,Manchester Airport M22,061 499 3320],[EuroDollar Rent-a-Car,Bridgeman Street,Bolton,0204 365373],[Europcar Car Rental,Flight Arrivals Hall/International Arrivals Hall,Manchester Airport M22,061 436 2200],[GB Silver Ladies Ltd,Outwood House 295 Wilmslow Road,Cheadle,Day: 061-436-1666/5560 Night: 0625-529763],[Grafton Car and Van Hire,Grafton Service Station 77 Chorley Old Road,Bolton,0204 383974],[HM Car and Van Hire,Unit 2G Hartford House Weston Road,Bolton,0204 380628],[Hertz,Arrivals Hall,Manchester Airport M22,061 437 8208],[Hertz,Manchester North branch,Bolton,061-273 8884],[Kenning Car and Van Rental,The Wilmslow Moathouse Oversley Ford Altrincham Road,Manchester Airport M22,0625 532240],[Kirkby Central Ltd,Manchester Road,Bolton,0204 389999/399797],[Majestic Car and Van Hire,St Helen's Road BL3 3PS,Bolton,0204 62335],[Mitchells Self Drive,Dawes Street,Bolton,0204 389988],[TJS (formerly AutoHire),135-137 Higher Bridge Street,Bolton,0204 394803]]]),20),carHireCompany(11)])

NewInfo:

[location(r,_20616)]

----- Entering Response Planner

----- Checking system obligations

Obligated goal:

1 know(s,[hireEvent(r,u,c,_17973),location(r,_20616)])

Goal 1 to be realised:

know(s,[hireEvent(r,u,c,_17973),location(r,_20616)])

----- Determining Central Concept

Possible system shifts:

[hireEvent(r,u,c,_17973)^coherent]

Central Concept: hireEvent(r,u,c,_17973)

----- Determining Propositional Content

Initial Agenda: [location(r,_22853)]

accurate: [location(r,_20616),hireEvent(r,u,c,_17973)]

valid: [location(r,_20616),hireEvent(r,u,c,_17973)]

consistent: [location(r,_20616),hireEvent(r,u,c,_17973)]

----- Ellipsis determined

Elliptical: ellipsis

```

Final Agenda: [location(r,_20616)]

is Accurate, Valid, Consistent, and Free from false implicatures

----- Leaving Response Planner

The semantic representation is:

interrog(_23596)
in(_23499,r,_20616)

----- Surface Generation:

Surface String:

Not yet connected to a surface generator.

|: 'In Entwistle'.

Parse:
[in(_24844,_24845,_24846),pname(_24846,entwistle),def(_24846),
num(_24846,sg),case(_24846,acc)]

Translation:
[location(_24845,_24846),pname(_24846,entwistle)]

User Goal:
[know(s,[location(r,entwistle),hireEvent(r,u,c,_25260),user(u),car(c)])]

Possible user shifts:
[car(c)^coherent,hireEvent(r,u,c,_25260)^cont]

Central Concept: hireEvent(r,u,c,_27129)
NewInfo: [nil(location(r,entwistle))]

Joint purpose:
[clarify(nil(location(r,entwistle)))]

Sys Goal:
1      clarify(know(s,[location(entwistle,_28619)]))

NewInfo:
[location(entwistle,_28612)]

----- Entering Response Planner

----- Checking system obligations

Obligated goal:
1      clarify(know(s,[location(entwistle,_28619)]))

Goal 1 to be realised:
clarify(know(s,[location(entwistle,_28619)]))

----- Determining Central Concept

```

```

----- Determining Propositional Content

Initial Agenda:  [location(entwistle,_29654)]

accurate: [location(entwistle,_28619)]

valid: [location(entwistle,_28619)]

consistent: [location(entwistle,_28619)]

----- Ellipsis determined

Elliptical: no_ellipsis

Final Agenda:  [location(entwistle,_28619)]

is Accurate, Valid, Consistent, and Free from false implicatures

----- Leaving Response Planner

The semantic representation is:

interrog(_30249)
in(_30193,entwistle,_28619)
be(b,_30249,entwistle)
stype(b,finf)
tense(b,pres-time)
num(b,sg)
pers(b,rd)

----- Surface Generation:

Surface String:

Not yet connected to a surface generator.

|: 'In Bolton'.

Parse:
[in(_30817,_30818,_30819),pname(_30819,bolton),def(_30819),
num(_30819,sg),case(_30819,acc)]

Translation:
[location(_30818,_30819),pname(_30819,bolton)]

User Goal:
[know(s,[location(entwistle,bolton),sublocation(bolton,entwistle)])]

Try accessible concepts for CC determination

Possible user shifts:
[pname(b,bolton)^awkward,location(e,b)^awkward,
sublocation(bolton,entwistle)^awkward]

Central Concept: location(_33034,entwistle)

```

NewInfo: [location(entwistle,bolton)]

Joint purpose:

[[carHireCompany,_35025,_35023,_35021,bolton,_35017],
[location(entwistle,bolton)]]

Sys Goal:

```
1      know(u,[cardinality(setOf(12,[[[ABC Car and Van Hire,Viola
Street,Bolton,0204 50075],[Boundary Garage 1972,Bury Road
BRIGHTMET,Bolton,0204 28607/35103],[British Car Rental,c/o Lex Mead
Manchester Road,Bolton,0204 31183],[Budget Rent-a-Car,Bradford House
287-289 Manchester Road,Bolton,0204 391611],[EuroDollar
Rent-a-Car,Bridgeman Street,Bolton,0204 365373],[Grafton Car and Van
Hire,Grafton Service Station 77 Chorley Old Road,Bolton,0204
383974],[HM Car and Van Hire,Unit 2G Hartford House Weston
Road,Bolton,0204 380628],[Hertz,Manchester North branch,Bolton,061-273
8884],[Kirkby Central Ltd,Manchester Road,Bolton,0204
389999/399797],[Majestic Car and Van Hire,St Helen's Road BL3
3PS,Bolton,0204 62335],[Mitchells Self Drive,Dawes Street,Bolton,0204
389988],[TJS (formerly AutoHire),135-137 Higher Bridge
Street,Bolton,0204
394803]]]),12),carHireCompany(12),location(entwistle,bolton))]
2      know(s,showall(13,carHireCompany(12)))
3      know(u,ypinfo(12,[[[ABC Car and Van Hire,Viola
Street,Bolton,0204 50075],[Boundary Garage 1972,Bury Road
BRIGHTMET,Bolton,0204 28607/35103],[British Car Rental,c/o Lex Mead
Manchester Road,Bolton,0204 31183],[Budget Rent-a-Car,Bradford House
287-289 Manchester Road,Bolton,0204 391611],[EuroDollar
Rent-a-Car,Bridgeman Street,Bolton,0204 365373],[Grafton Car and Van
Hire,Grafton Service Station 77 Chorley Old Road,Bolton,0204
383974],[HM Car and Van Hire,Unit 2G Hartford House Weston
Road,Bolton,0204 380628],[Hertz,Manchester North branch,Bolton,061-273
8884],[Kirkby Central Ltd,Manchester Road,Bolton,0204
389999/399797],[Majestic Car and Van Hire,St Helen's Road BL3
3PS,Bolton,0204 62335],[Mitchells Self Drive,Dawes Street,Bolton,0204
389988],[TJS (formerly AutoHire),135-137 Higher Bridge
Street,Bolton,0204 394803]]]))
```

NewInfo:

```
[cardinality(setOf(12,[[[ABC Car and Van Hire,Viola Street,Bolton,0204
50075],[Boundary Garage 1972,Bury Road BRIGHTMET,Bolton,0204
28607/35103],[British Car Rental,c/o Lex Mead Manchester
Road,Bolton,0204 31183],[Budget Rent-a-Car,Bradford House 287-289
Manchester Road,Bolton,0204 391611],[EuroDollar Rent-a-Car,Bridgeman
Street,Bolton,0204 365373],[Grafton Car and Van Hire,Grafton Service
Station 77 Chorley Old Road,Bolton,0204 383974],[HM Car and Van
Hire,Unit 2G Hartford House Weston Road,Bolton,0204
380628],[Hertz,Manchester North branch,Bolton,061-273 8884],[Kirkby
Central Ltd,Manchester Road,Bolton,0204 389999/399797],[Majestic Car
and Van Hire,St Helen's Road BL3 3PS,Bolton,0204 62335],[Mitchells
Self Drive,Dawes Street,Bolton,0204 389988],[TJS (formerly
AutoHire),135-137 Higher Bridge Street,Bolton,0204
394803]]]),12),ypinfo(12,[[[ABC Car and Van Hire,Viola
Street,Bolton,0204 50075],[Boundary Garage 1972,Bury Road
BRIGHTMET,Bolton,0204 28607/35103],[British Car Rental,c/o Lex Mead
Manchester Road,Bolton,0204 31183],[Budget Rent-a-Car,Bradford House
```

287-289 Manchester Road,Bolton,0204 391611],[EuroDollar
Rent-a-Car,Bridgeman Street,Bolton,0204 365373],[Grafton Car and Van
Hire,Grafton Service Station 77 Chorley Old Road,Bolton,0204
383974],[HM Car and Van Hire,Unit 2G Hartford House Weston
Road,Bolton,0204 380628],[Hertz,Manchester North branch,Bolton,061-273
8884],[Kirkby Central Ltd,Manchester Road,Bolton,0204
389999/399797],[Majestic Car and Van Hire,St Helen's Road BL3
3PS,Bolton,0204 62335],[Mitchells Self Drive,Dawes Street,Bolton,0204
389988],[TJS (formerly AutoHire),135-137 Higher Bridge
Street,Bolton,0204 394803]]]]]

----- Entering Response Planner

----- Checking system obligations

Obligated goal:

```
1      know(u,[cardinality(setOf(12,[[[ABC Car and Van Hire,Viola
Street,Bolton,0204 50075],[Boundary Garage 1972,Bury Road
Brightmet,Bolton,0204 28607/35103],[British Car Rental,c/o Lex Mead
Manchester Road,Bolton,0204 31183],[Budget Rent-a-Car,Bradford House
287-289 Manchester Road,Bolton,0204 391611],[EuroDollar
Rent-a-Car,Bridgeman Street,Bolton,0204 365373],[Grafton Car and Van
Hire,Grafton Service Station 77 Chorley Old Road,Bolton,0204
383974],[HM Car and Van Hire,Unit 2G Hartford House Weston
Road,Bolton,0204 380628],[Hertz,Manchester North branch,Bolton,061-273
8884],[Kirkby Central Ltd,Manchester Road,Bolton,0204
389999/399797],[Majestic Car and Van Hire,St Helen's Road BL3
3PS,Bolton,0204 62335],[Mitchells Self Drive,Dawes Street,Bolton,0204
389988],[TJS (formerly AutoHire),135-137 Higher Bridge
Street,Bolton,0204
394803]]]]),12),carHireCompany(12),location(entwistle,bolton))]
2      know(s,showall(13,carHireCompany(12)))
```

Goal 1 to be realised:

```
know(u,[cardinality(setOf(12,[[[ABC Car and Van Hire,Viola
Street,Bolton,0204 50075],[Boundary Garage 1972,Bury Road
Brightmet,Bolton,0204 28607/35103],[British Car Rental,c/o Lex Mead
Manchester Road,Bolton,0204 31183],[Budget Rent-a-Car,Bradford House
287-289 Manchester Road,Bolton,0204 391611],[EuroDollar
Rent-a-Car,Bridgeman Street,Bolton,0204 365373],[Grafton Car and Van
Hire,Grafton Service Station 77 Chorley Old Road,Bolton,0204
383974],[HM Car and Van Hire,Unit 2G Hartford House Weston
Road,Bolton,0204 380628],[Hertz,Manchester North branch,Bolton,061-273
8884],[Kirkby Central Ltd,Manchester Road,Bolton,0204
389999/399797],[Majestic Car and Van Hire,St Helen's Road BL3
3PS,Bolton,0204 62335],[Mitchells Self Drive,Dawes Street,Bolton,0204
389988],[TJS (formerly AutoHire),135-137 Higher Bridge
Street,Bolton,0204
394803]]]]),12),carHireCompany(12),location(entwistle,bolton))]
```

----- Determining Central Concept

----- Determining Propositional Content

Initial Agenda: [cardinality(setOf(12,[[[ABC Car and Van Hire,Viola

Street,Bolton,0204 50075],[Boundary Garage 1972,Bury Road
 Brightmet,Bolton,0204 28607/35103],[British Car Rental,c/o Lex Mead
 Manchester Road,Bolton,0204 31183],[Budget Rent-a-Car,Bradford House
 287-289 Manchester Road,Bolton,0204 391611],[EuroDollar
 Rent-a-Car,Bridgeman Street,Bolton,0204 365373],[Grafton Car and Van
 Hire,Grafton Service Station 77 Chorley Old Road,Bolton,0204
 383974],[HM Car and Van Hire,Unit 2G Hartford House Weston
 Road,Bolton,0204 380628],[Hertz,Manchester North branch,Bolton,061-273
 8884],[Kirkby Central Ltd,Manchester Road,Bolton,0204
 389999/399797],[Majestic Car and Van Hire,St Helen's Road BL3
 3PS,Bolton,0204 62335],[Mitchells Self Drive,Dawes Street,Bolton,0204
 389988],[TJS (formerly AutoHire),135-137 Higher Bridge
 Street,Bolton,0204 394803]])),12),ypinfo(12,[[[ABC Car and Van
 Hire,Viola Street,Bolton,0204 50075],[Boundary Garage 1972,Bury Road
 Brightmet,Bolton,0204 28607/35103],[British Car Rental,c/o Lex Mead
 Manchester Road,Bolton,0204 31183],[Budget Rent-a-Car,Bradford House
 287-289 Manchester Road,Bolton,0204 391611],[EuroDollar
 Rent-a-Car,Bridgeman Street,Bolton,0204 365373],[Grafton Car and Van
 Hire,Grafton Service Station 77 Chorley Old Road,Bolton,0204
 383974],[HM Car and Van Hire,Unit 2G Hartford House Weston
 Road,Bolton,0204 380628],[Hertz,Manchester North branch,Bolton,061-273
 8884],[Kirkby Central Ltd,Manchester Road,Bolton,0204
 389999/399797],[Majestic Car and Van Hire,St Helen's Road BL3
 3PS,Bolton,0204 62335],[Mitchells Self Drive,Dawes Street,Bolton,0204
 389988],[TJS (formerly AutoHire),135-137 Higher Bridge
 Street,Bolton,0204 394803]]))]

accurate: [cardinality(setOf(12,[[[ABC Car and Van Hire,Viola
 Street,Bolton,0204 50075],[Boundary Garage 1972,Bury Road
 Brightmet,Bolton,0204 28607/35103],[British Car Rental,c/o Lex Mead
 Manchester Road,Bolton,0204 31183],[Budget Rent-a-Car,Bradford House
 287-289 Manchester Road,Bolton,0204 391611],[EuroDollar
 Rent-a-Car,Bridgeman Street,Bolton,0204 365373],[Grafton Car and Van
 Hire,Grafton Service Station 77 Chorley Old Road,Bolton,0204
 383974],[HM Car and Van Hire,Unit 2G Hartford House Weston
 Road,Bolton,0204 380628],[Hertz,Manchester North branch,Bolton,061-273
 8884],[Kirkby Central Ltd,Manchester Road,Bolton,0204
 389999/399797],[Majestic Car and Van Hire,St Helen's Road BL3
 3PS,Bolton,0204 62335],[Mitchells Self Drive,Dawes Street,Bolton,0204
 389988],[TJS (formerly AutoHire),135-137 Higher Bridge
 Street,Bolton,0204 394803]])),12),ypinfo(12,[[[ABC Car and Van
 Hire,Viola Street,Bolton,0204 50075],[Boundary Garage 1972,Bury Road
 Brightmet,Bolton,0204 28607/35103],[British Car Rental,c/o Lex Mead
 Manchester Road,Bolton,0204 31183],[Budget Rent-a-Car,Bradford House
 287-289 Manchester Road,Bolton,0204 391611],[EuroDollar
 Rent-a-Car,Bridgeman Street,Bolton,0204 365373],[Grafton Car and Van
 Hire,Grafton Service Station 77 Chorley Old Road,Bolton,0204
 383974],[HM Car and Van Hire,Unit 2G Hartford House Weston
 Road,Bolton,0204 380628],[Hertz,Manchester North branch,Bolton,061-273
 8884],[Kirkby Central Ltd,Manchester Road,Bolton,0204
 389999/399797],[Majestic Car and Van Hire,St Helen's Road BL3
 3PS,Bolton,0204 62335],[Mitchells Self Drive,Dawes Street,Bolton,0204
 389988],[TJS (formerly AutoHire),135-137 Higher Bridge
 Street,Bolton,0204
 394803]])),carHireCompany(12),location(entwistle,bolton)]

```

valid: [cardinality(setOf(12,[[[ABC Car and Van Hire,Viola
Street,Bolton,0204 50075],[Boundary Garage 1972,Bury Road
Brechtmet,Bolton,0204 28607/35103],[British Car Rental,c/o Lex Mead
Manchester Road,Bolton,0204 31183],[Budget Rent-a-Car,Bradford House
287-289 Manchester Road,Bolton,0204 391611],[EuroDollar
Rent-a-Car,Bridgeman Street,Bolton,0204 365373],[Grafton Car and Van
Hire,Grafton Service Station 77 Chorley Old Road,Bolton,0204
383974],[HM Car and Van Hire,Unit 2G Hartford House Weston
Road,Bolton,0204 380628],[Hertz,Manchester North branch,Bolton,061-273
8884],[Kirkby Central Ltd,Manchester Road,Bolton,0204
389999/399797],[Majestic Car and Van Hire,St Helen's Road BL3
3PS,Bolton,0204 62335],[Mitchells Self Drive,Dawes Street,Bolton,0204
389988],[TJS (formerly AutoHire),135-137 Higher Bridge
Street,Bolton,0204 394803]])),12),ypinfo(12,[[[ABC Car and Van
Hire,Viola Street,Bolton,0204 50075],[Boundary Garage 1972,Bury Road
Brechtmet,Bolton,0204 28607/35103],[British Car Rental,c/o Lex Mead
Manchester Road,Bolton,0204 31183],[Budget Rent-a-Car,Bradford House
287-289 Manchester Road,Bolton,0204 391611],[EuroDollar
Rent-a-Car,Bridgeman Street,Bolton,0204 365373],[Grafton Car and Van
Hire,Grafton Service Station 77 Chorley Old Road,Bolton,0204
383974],[HM Car and Van Hire,Unit 2G Hartford House Weston
Road,Bolton,0204 380628],[Hertz,Manchester North branch,Bolton,061-273
8884],[Kirkby Central Ltd,Manchester Road,Bolton,0204
389999/399797],[Majestic Car and Van Hire,St Helen's Road BL3
3PS,Bolton,0204 62335],[Mitchells Self Drive,Dawes Street,Bolton,0204
389988],[TJS (formerly AutoHire),135-137 Higher Bridge
Street,Bolton,0204
394803]])),carHireCompany(12),location(entwistle,bolton)]

```

```

consistent: [cardinality(setOf(12,[[[ABC Car and Van Hire,Viola
Street,Bolton,0204 50075],[Boundary Garage 1972,Bury Road
Brechtmet,Bolton,0204 28607/35103],[British Car Rental,c/o Lex Mead
Manchester Road,Bolton,0204 31183],[Budget Rent-a-Car,Bradford House
287-289 Manchester Road,Bolton,0204 391611],[EuroDollar
Rent-a-Car,Bridgeman Street,Bolton,0204 365373],[Grafton Car and Van
Hire,Grafton Service Station 77 Chorley Old Road,Bolton,0204
383974],[HM Car and Van Hire,Unit 2G Hartford House Weston
Road,Bolton,0204 380628],[Hertz,Manchester North branch,Bolton,061-273
8884],[Kirkby Central Ltd,Manchester Road,Bolton,0204
389999/399797],[Majestic Car and Van Hire,St Helen's Road BL3
3PS,Bolton,0204 62335],[Mitchells Self Drive,Dawes Street,Bolton,0204
389988],[TJS (formerly AutoHire),135-137 Higher Bridge
Street,Bolton,0204 394803]])),12),ypinfo(12,[[[ABC Car and Van
Hire,Viola Street,Bolton,0204 50075],[Boundary Garage 1972,Bury Road
Brechtmet,Bolton,0204 28607/35103],[British Car Rental,c/o Lex Mead
Manchester Road,Bolton,0204 31183],[Budget Rent-a-Car,Bradford House
287-289 Manchester Road,Bolton,0204 391611],[EuroDollar
Rent-a-Car,Bridgeman Street,Bolton,0204 365373],[Grafton Car and Van
Hire,Grafton Service Station 77 Chorley Old Road,Bolton,0204
383974],[HM Car and Van Hire,Unit 2G Hartford House Weston
Road,Bolton,0204 380628],[Hertz,Manchester North branch,Bolton,061-273
8884],[Kirkby Central Ltd,Manchester Road,Bolton,0204
389999/399797],[Majestic Car and Van Hire,St Helen's Road BL3
3PS,Bolton,0204 62335],[Mitchells Self Drive,Dawes Street,Bolton,0204
389988],[TJS (formerly AutoHire),135-137 Higher Bridge
Street,Bolton,0204

```


394803]]]),carHireCompany(12),location(entwistle,bolton))

----- Ellipsis determined

Elliptical: no_ellipsis

Final Agenda: [cardinality(setOf(12,[[[ABC Car and Van Hire,Viola Street,Bolton,0204 50075],[E

is Accurate, Valid, Consistent, and Free from false implicatures

Goal 2 to be realised:

know(s,showall(13,carHireCompany(12)))

----- Determining Central Concept

----- Determining Propositional Content

Initial Agenda: [final,showall(13,carHireCompany(12))]

----- Ellipsis determined

Elliptical: no_ellipsis

Final Agenda: [showall,carHireCompany(12)]

is Accurate, Valid, Consistent, and Free from false implicatures

----- Leaving Response Planner

The semantic representation is:

12

[[[ABC Car and Van Hire,Viola Street,Bolton,0204 50075],[Boundary
Garage 1972,Bury Road Broughton,Bolton,0204 28607/35103],[British Car
Rental,c/o Lex Mead Manchester Road,Bolton,0204 31183],[Budget
Rent-a-Car,Bradford House 287-289 Manchester Road,Bolton,0204
391611],[EuroDollar Rent-a-Car,Bridgeman Street,Bolton,0204
365373],[Grafton Car and Van Hire,Grafton Service Station 77 Chorley
Old Road,Bolton,0204 383974],[HM Car and Van Hire,Unit 2G Hartford
House Weston Road,Bolton,0204 380628],[Hertz,Manchester North
branch,Bolton,061-273 8884],[Kirkby Central Ltd,Manchester
Road,Bolton,0204 389999/399797],[Majestic Car and Van Hire,St Helen's
Road BL3 3PS,Bolton,0204 62335],[Mitchells Self Drive,Dawes
Street,Bolton,0204 389988],[TJS (formerly AutoHire),135-137 Higher
Bridge Street,Bolton,0204 394803]]]

interrog(_41423)

you(s)

num(s,sg)

case(s,nom)

want(_41000,s,_41002)

stype(_41000,finf)

tense(_41000,pres-time)

num(_41000,sg)

pers(_41000,st)

stype(_41000,finf)

```
tense(_41000,pres-time)
num(_41000,sg)
pers(_41000,nd)
see(_41002,s,12)
stype(_41002,infinf)
forall(12)
num(12,pl)
pers(12,rd)
case(12,acc)
carhirecompany(12)
```

----- Surface Generation:

Surface String:

12

```
interrog(_41423) you(s) num(s,sg) case(s,nom) want(_41000,s,_41002)
stype(_41000,finf) tense(_41000,pres-time) num(_41000,sg)
pers(_41000,st) stype(_41000,finf) tense(_41000,pres-time)
num(_41000,sg) pers(_41000,nd) see(_41002,s,12) stype(_41002,infinf)
forall(12) num(12,pl) pers(12,rd) case(12,acc) carhirecompany(12)
```

|: 'Are there any vegetarian restaurants in Manchester'.

Parse:

```
[interrog(_46307),existq(_46303),forsome(_46303),vegetarian(_46294,_46303),
basef(_46294),restaurant(_46303),num(_46303,pl),case(_46303,nom),
kind(_46303,countn),in(_46265,_46303,_46267),pp-def(_46265),
pname(_46267,manchester),def(_46267),num(_46267,sg),case(_46267,acc)]
```

Translation:

```
[existq(_46303),typeof(_46303,vegetarian),restaurant(_46303),
location(_46303,_46267),pname(_46267,manchester)]
```

User Goal:

```
[know(u,[existq(re),typeof(re,vegetarian),restaurant(re),
location(re,manchester)])]
```

Possible user shifts:

```
[restaurant(re)^awkward]
```

Central Concept: restaurant(re)

NewInfo: [existq(re)]

Joint purpose:

```
[existq(re),typeof(re,vegetarian),restaurant(re),location(re,manchester)]
```

AM headings: [restaurant]

Query: [restaurant,vegetarian,_50800,_50798,manchester,_50794]

Specs: [existq(re),typeof(re,vegetarian),location(re,manchester)]

Sys Goal:

```
1      know(u,supplyinfo([restaurant(re),existq(re),typeof(re,vegetarian),
```

```

location(re,manchester)))
2      know(u,[cardinality(setOf(re,[[[VEGETARIAN,Billies Vegetarian
Restaurant Fully Licensed/Organic Wines A Speciality,115 Manchester
Rd,Manchester 21,061-881 93],[VEGETARIAN,On the Eight Day,111 Oxford
Rd,Manchester 1,061-273 1850]]]),2),
restaurant(re),existq(re),typeof(re,vegetarian),location(re,manchester))]
3      know(u,ypinfo(re,[[[VEGETARIAN,Billies Vegetarian Restaurant
Fully Licensed/Organic Wines A Speciality,115 Manchester Rd,Manchester
21,061-881 93],[VEGETARIAN,On the Eight Day,111 Oxford Rd,Manchester
1,061-273 1850]]]))

```

NewInfo:

```

[ypinfo(re,[[[VEGETARIAN,Billies Vegetarian Restaurant Fully
Licensed/Organic Wines A Speciality,115 Manchester Rd,Manchester
21,061-881 93],[VEGETARIAN,On the Eight Day,111 Oxford Rd,Manchester
1,061-273 1850]]])]

```

----- Entering Response Planner

----- Checking system obligations

Obligated goal:

```

1      know(u,supplyinfo([restaurant(re),existq(re),
typeof(re,vegetarian),location(re,manchester)]))
2      know(u,[cardinality(setOf(re,[[[VEGETARIAN,Billies Vegetarian
Restaurant Fully Licensed/Organic Wines A Speciality,115 Manchester
Rd,Manchester 21,061-881 93],[VEGETARIAN,On the Eight Day,111 Oxford
Rd,Manchester 1,061-273 1850]]]),2),
restaurant(re),existq(re),typeof(re,vegetarian),location(re,manchester))]

```

Goal 1 to be realised:

```

know(u,supplyinfo([restaurant(re),existq(re),
typeof(re,vegetarian),location(re,manchester)]))

```

----- Determining Central Concept

----- Determining Propositional Content

```

Initial Agenda: [final,supplyinfo([restaurant(re),existq(re),
typeof(re,vegetarian),location(re,manchester)])]

```

----- Ellipsis determined

Elliptical: no_ellipsis

```

Final Agenda: [supplyinfo,[restaurant(re),existq(re),
typeof(re,vegetarian),location(re,manchester)]]

```

is Accurate, Valid, Consistent, and Free from false implicatures

Goal 2 to be realised:

```

know(u,[cardinality(setOf(re,[[[VEGETARIAN,Billies Vegetarian
Restaurant Fully Licensed/Organic Wines A Speciality,115 Manchester
Rd,Manchester 21,061-881 93],[VEGETARIAN,On the Eight Day,111 Oxford
Rd,Manchester 1,061-273 1850]]]),2),

```

restaurant(re),existq(re),typeof(re,vegetarian),location(re,manchester))]

----- Determining Central Concept

----- Determining Propositional Content

Initial Agenda: [ypinfo(re,[[[VEGETARIAN,Billies Vegetarian Restaurant Fully Licensed/Organic Wines A Speciality,115 Manchester Rd,Manchester 21,061-881 93],[VEGETARIAN,On the Eight Day,111 Oxford Rd,Manchester 1,061-273 1850]]]])]

accurate: [ypinfo(re,[[[VEGETARIAN,Billies Vegetarian Restaurant Fully Licensed/Organic Wines A Speciality,115 Manchester Rd,Manchester 21,061-881 93],[VEGETARIAN,On the Eight Day,111 Oxford Rd,Manchester 1,061-273 1850]]]),cardinality(setOf(re,[[[VEGETARIAN,Billies Vegetarian Restaurant Fully Licensed/Organic Wines A Speciality,115 Manchester Rd,Manchester 21,061-881 93],[VEGETARIAN,On the Eight Day,111 Oxford Rd,Manchester 1,061-273 1850]]]),2),restaurant(re),existq(re),typeof(re,vegetarian),location(re,manchester)]

valid: [ypinfo(re,[[[VEGETARIAN,Billies Vegetarian Restaurant Fully Licensed/Organic Wines A Speciality,115 Manchester Rd,Manchester 21,061-881 93],[VEGETARIAN,On the Eight Day,111 Oxford Rd,Manchester 1,061-273 1850]]]),cardinality(setOf(re,[[[VEGETARIAN,Billies Vegetarian Restaurant Fully Licensed/Organic Wines A Speciality,115 Manchester Rd,Manchester 21,061-881 93],[VEGETARIAN,On the Eight Day,111 Oxford Rd,Manchester 1,061-273 1850]]]),2),restaurant(re),existq(re),typeof(re,vegetarian),location(re,manchester)]

consistent: [ypinfo(re,[[[VEGETARIAN,Billies Vegetarian Restaurant Fully Licensed/Organic Wines A Speciality,115 Manchester Rd,Manchester 21,061-881 93],[VEGETARIAN,On the Eight Day,111 Oxford Rd,Manchester 1,061-273 1850]]]),cardinality(setOf(re,[[[VEGETARIAN,Billies Vegetarian Restaurant Fully Licensed/Organic Wines A Speciality,115 Manchester Rd,Manchester 21,061-881 93],[VEGETARIAN,On the Eight Day,111 Oxford Rd,Manchester 1,061-273 1850]]]),2),restaurant(re),existq(re),typeof(re,vegetarian),location(re,manchester)]

----- Ellipsis determined

Elliptical: no_ellipsis

Final Agenda: [ypinfo(re,[[[VEGETARIAN,Billies Vegetarian Restaurant Fully Licensed/Organic Wines A Speciality,115 Manchester Rd,Manchester 21,061-881 93],[VEGETARIAN,On the Eight Day,111 Oxford Rd,Manchester 1,061-273 1850]]]),cardinality(setOf(re,[[[VEGETARIAN,Billies Vegetarian Restaurant Fully Licensed/Organic Wines A Speciality,115 Manchester Rd,Manchester 21,061-881 93],[VEGETARIAN,On the Eight Day,111 Oxford Rd,Manchester 1,061-273 1850]]]),2),restaurant(re),existq(re),typeof(re,vegetarian),location(re,manchester)]

is Accurate, Valid, Consistent, and Free from false implicatures

----- Leaving Response Planner

The semantic representation is:

```

here(_55818)
come(_55813,_55814)
stype(_55813,finf)
tense(_55813,pres-time)
num(_55813,sg)
pers(_55813,rd)
def(_55814)
list(_55814)
num(_55814,sg)
case(_55814,acc)
kind(_55814,countn)
of(_55761,_55814,re)
def(re)
num(re,pl)
case(re,acc)
restaurant(re)
existq(re)
vegetarian(_55914,re)
in(_55945,re,manchester)

```

VEGETARIAN

Billies Vegetarian Restaurant Fully Licensed/Organic Wines A Speciality
 115 Manchester Rd
 Manchester 21
 061-881 93

VEGETARIAN

On the Eight Day
 111 Oxford Rd
 Manchester 1
 061-273 1850

----- Surface Generation:

Surface String:

Here comes the list:

VEGETARIAN

Billies Vegetarian Restaurant Fully Licensed/Organic Wines A Speciality
 115 Manchester Rd
 Manchester 21
 061-881 93

VEGETARIAN

On the Eight Day
 111 Oxford Rd
 Manchester 1
 061-273 1850

|: hei.

Parse:

hei

Ok. Bye then.

no

| ?- ^D\$

22-skye> ^D\$exit

script done on Sun Jul 17 12:01:41 1994