January 1992

# Planning Responses From High-Level Goals: Adopting the Respondent's Perspective Cooperative Response Generation

Brant Cheikes
*University of Pennsylvania*



University of Pennsylvania Department of Computer and Information Science Technical Report No. MS-CIS-92-14.

# Planning Responses From High-Level Goals: Adopting the Respondent's Perspective Cooperative Response Generation

## Abstract

Within the natural-language research community it has long been acknowledged that the conventions and pragmatics of natural-language communication often oblige dialogue systems to consider and address the underlying purposes of queries in their responses rather than answering them literally and without further comment or elaboration. Such systems cannot simply translate their users' requests into transactions on database or expert systems, but must apply many more complex reasoning mechanisms to the task of selecting responses that are both appropriate and useful. This idea has given rise to a broadly-defined program of research in *cooperative response generation* (CRG).

Research in CRG carried on over more than a decade has yielded a substantial body of literature. Analysis of that literature, however, shows that investigators have focused primarily on modeling *manifestations* of cooperative behavior without directly considering the nature and motivations of the behavior itself. But if we want to develop natural language dialogue systems that are truly to function as *cooperative respondents* instead of serving only as models of particular kinds of cooperative responses, a different approach is required.

I identify two opposing perspectives on the process of cooperative response generation: the *questioner-based* and the *respondent-based* perspectives. I argue that past research efforts have largely been questioner-based, and that this view has led to the development of theories that are incompatible and cannot be integrated. I propose the respondent-based view as an alternative, and provide evidence that taking such a perspective might allow several interesting but otherwise poorly-understood aspects of cooperative response behavior to be modeled.

The final portion of the dissertation explores the computational implications of a respondent-based perspective. I outline the architecture of a *Cooperative Response Planning System*, a dialogue system that raises, reasons about, and attempts to satisfy high-level cooperative goals in its responses. This architecture constitutes a first approximation to a theory of how a system might reason from the beliefs it derives from a questioner's utterances to choose a cooperative response. The processing of two sample responses in this framework is described in detail to illustrate the architecture's capabilities.

## Comments

University of Pennsylvania Department of Computer and Information Science Technical Report No. MS-CIS-92-14.

# Planning Responses From High-Level Goals: Adopting The Respondent's Perspective Cooperative Response Generation

MS-CIS-92-14
LINC LAB 215

Brant Cheikes

University of Pennsylvania
School of Engineering and Applied Science
Computer and Information Science Department
Philadelphia, PA 19104-6389

February 1992

# Planning Responses from High-Level Goals: Adopting the Respondent's Perspective in Cooperative Response Generation

Brant A. Cheikes
University of Pennsylvania
Department of Computer and Information Science
Philadelphia, PA 19104-6389
brant@linc.cis.upenn.edu

January 27, 1992

## Abstract

Within the natural-language research community it has long been acknowledged that the conventions and pragmatics of natural-language communication often oblige dialogue systems to consider and address the underlying purposes of queries in their responses rather than answering them literally and without further comment or elaboration. Such systems cannot simply translate their users' requests into transactions on database or expert systems, but must apply many more complex reasoning mechanisms to the task of selecting responses that are both appropriate and useful. This idea has given rise to a broadly-defined program of research in *cooperative response generation* (CRG).

Research in CRG carried on over more than a decade has yielded a substantial body of literature. Analysis of that literature, however, shows that investigators have focused primarily on modeling *manifestations* of cooperative behavior without directly considering the nature and motivations of the behavior itself. But if we want to develop natural-language dialogue systems that are truly to function as *cooperative respondents* instead of serving only as models of particular kinds of cooperative responses, a different approach is required.

I identify two opposing perspectives on the process of cooperative response generation: the **questioner-based** and the **respondent-based** perspectives. I argue that past research efforts have largely been questioner-based, and that this view has led to the development of theories that are incompatible and cannot be integrated. I propose the respondent-based view as an alternative, and provide evidence that taking such a perspective might allow several interesting but otherwise poorly-understood aspects of cooperative response behavior to be modeled.

The final portion of the dissertation explores the computational implications of a respondent-based perspective. I outline the architecture of a **Cooperative Response Planning System**, a dialogue system that raises, reasons about, and attempts to satisfy high-level cooperative goals in its responses. This architecture constitutes a first approximation to a theory of how a system might reason from the beliefs it derives from a questioner's utterances to choose a cooperative response. The processing of two sample responses in this framework is described in detail to illustrate the architecture's capabilities.

# Contents

# List of Figures

# Chapter 1

# Introduction

This dissertation focuses on the design of natural-language dialogue systems that are to be conceived of as cooperative respondents. A distinction between *questioner-based* and *respondent-based* perspectives on cooperative response behavior is developed, and the implications of the latter framework for dialogue system design are explored.

## 1.1 Cooperative Response Generation

Within the natural-language research community it has long been acknowledged that the conventions and pragmatics of natural-language communication often oblige natural-language dialogue systems to consider and address in their responses the underlying purposes of queries rather than answering them literally and without further comment or elaboration. These systems cannot simply translate their users' queries into transactions on database or expert systems, but must apply many more complex reasoning mechanisms to the task of selecting appropriate and useful responses. Researchers have argued that if question-answering systems are to be perceived by their users as acceptable dialogue partners, they must be able to provide cooperative responses (also called *extended responses* [Wahlster 83, Finin 86, Webber 86]) that may include such general elements as:

- the so-called *direct answer*: the information literally requested;
- information or action related to the direct answer;
- information or action *in place of* the direct answer;
- information or action that is related to one or more of the questioner's stated, inferred, or assumed goals.

The idea that responses may need to address concerns that are well beyond the explicit request has spawned a broadly-defined program of research in *cooperative response generation* (CRG).

Research in CRG over more than a decade has yielded a substantial body of results. Many of these efforts have been based on analyses of examples found in transcripts of naturally occurring dialogue. A commonly-employed research approach follows these steps:

- identifying forms or patterns of response behavior that seem to follow from principles of cooperation,
- characterizing the properties of those responses that correlate with people's judgments of their cooperativeness,
- circumscribing the conditions under which those responses are typically produced, and
- developing computational methods that dialogue systems can employ to produce (or to support the production of) responses having the desired properties in the predefined situations.

The research to be presented in this dissertation is not an addition in kind, but rather is set against a backdrop composed of this body of accumulated results. I will argue that when we evaluate past efforts in light of the possible long-term goals of CRG research, it becomes less than clear whether any real progress has been achieved.

### 1.1.1 Long-range research goals

There are two long-range goals toward which research in CRG might be directed. One is a goal of engineering, the other a goal of science. The engineering goal is to develop computer technology and algorithms that enable dialogue systems to respond to requests in ways that their users perceive to be satisfactorily informative, clear, and helpful. Researchers have taken steps in this direction by specifying inference procedures that support different kinds of cooperative response decisions, characterizing different knowledge sources that dialogue systems draw upon in the process of computing their responses, and identifying properties of different applications and/or domains of discourse that affect the kinds of cooperative responses that must be produced. When evaluating engineering solutions, we typically apply such criteria as domain independence, computational feasibility, and scalability.

The scientific goal is to identify the underlying principles and cognitive mechanisms of cooperation and cooperative discourse. As a scientific enterprise, the study of CRG leads us to investigate questions of a more philosophical kind:

- What are the characteristics that define a "cooperative" response?
- What does it mean for one agent to cooperate with another?
- How does cooperation among agents affect their behavior in a dialogue?
- What are the general principles that drive cooperative behavior?

Scientific results are typically evaluated in terms of their breadth of coverage of naturally occurring data and the degree to which they accord with known theories of cognition.

As part of this dissertation I will show that past work on CRG has been directed almost exclusively toward the engineering goal, while the scientific goal has received scant attention. It has been implicitly assumed that these goals can be pursued independently; however, two conclusions we will draw from the present research are that (1) progress in the engineering of cooperative response behavior has not significantly increased scientific understanding of cooperation, and (2) without scientific progress, our engineering efforts will be limited to the development of systems that are able only to reproduce fixed patterns of cooperative response behavior, rather than to reason and act as true cooperative dialogue partners.

### 1.1.2 Background: the integration problem

As originally conceived, this research project was an attempt to take the next logical step toward the engineering goal by addressing what I have previously called the *integration problem* [Cheikes 87]. This is the problem of combining restricted models of cooperative responses to form more capable dialogue systems.

We have the intuition that cooperation manifests itself in language in many ways, both subtle and blatant. In the past, researchers have studied cooperative response generation largely by characterizing regular patterns of "helpful" question-answering behavior, then developing computational techniques that could be employed to reproduce those patterns. There seems to have been a presumption that enough research of this kind would lead to the development of dialogue systems able to demonstrate a realistic range of cooperative response ability, a range approximating that of human cooperative respondents. Gal expressed this presumption most succinctly in the introduction to her doctoral thesis [Gal 88, p. 23]:

> All [approaches to cooperative question answering] are generally complementary to each other, since they each solve particular problems, among the numerous ones which must be faced in building a truly cooperative interface—that is, an interface cooperative enough to allow the user to forget that he is questioning a machine.

This comment suggests a view in which researchers are solving important subproblems along the route to constructing "truly cooperative" interfaces. But after a review of numerous systems that all purported to generate "cooperative responses," it was not obvious how (or if) the different problems and proposed solutions related to one another and to the design of "truly cooperative" interfaces. I thus set out to investigate this question: how could the results of past research be *integrated* to form dialogue systems possessing a repertoire of cooperative talents?

### 1.1.3 Difficulties encountered

Research up to the time at which this project commenced had generally treated the study of CRG as a process of naming and modeling distinct forms of cooperative response behavior. Kaplan, for example, had designed a system that generated what he called "corrective indirect responses" when it detected an extensional query failure [Kaplan 82]. Mays had investigated both corrective responses to intensional query failures [Mays 80] and monitor offers on dynamic databases [Mays 84b]. Various kinds of misconception-correction strategies had been considered [McCoy 85, Quilici 88], and Allen had developed a model of plan inference to support "obstacle elimination" [Allen 83]. These are just a few examples; we will see more in the next chapter.

Given the foundation that all this research provided, the most obvious approach to achieving integration at the time was to design a *blackboard system* [Nii 86b] in which the different theories of cooperative question answering would be implemented as modules of some sort. Early efforts along this line are discussed in [Cheikes 87] and [Cheikes 89].

After several unsatisfactory attempts to define the architecture of a blackboard-based cooperative response generation system, it became clear that there were subtle difficulties with the way CRG research had been approached, the effect being that the accumulated results resisted integration efforts. The integration work foundered for several reasons:

- Research projects had been conducted in isolation from one another, and thus they lacked a common conceptual framework.
- Different assumptions had been made about what information was available to the system (in terms of both what was contained in systems' general knowledge bases and what knowledge was derived from user input).
- Each system assumed that it was responsible for producing the entire response to the user's input in some particular dialogue context.
- The cooperativeness of many response forms seemed to depend on unstated assumptions about the goals that the user was pursuing.

The conclusion that I drew was that the various systems, when considered individually, seemed to neatly solve some useful problems along the way to "truly cooperative" dialogue systems, but, when considered as complementary efforts leading toward some common end, represented incompatible theories.

This conclusion raised a new crop of discomfiting questions:

- If previous efforts were not leading to "truly cooperative" dialogue systems, where were they leading?
- Why were the results of those efforts so incompatible?
- What changes of approach were called for in order to make progress in cooperative question answering?

It seemed that the whole approach to studying and modeling cooperative response behavior required complete rethinking. This dissertation presents the results to date of my reconsideration of the CRG problem.

## 1.2 Summary of Research

My rethinking of the problem of cooperative response generation proceeded through four stages. First, I returned to the literature and undertook an in-depth analysis and critique. That work is reported in Chapter 2. Next, I developed an abstract characterization of past work that explained the conclusions of my literature critique. This characterization is called the *questioner-based perspective*, and is described in Section 3.1. The insights provided by this characterization, together with my analyses of examples of naturally occurring request-response pairs (described in Section 1.3), led to the development of a new theoretical perspective on the process of cooperative question answering, the *respondent-based perspective.* This new perspective is described in Section 3.2. Finally, I explored the consequences of the respondent-based perspective by sketching an architecture design for respondent-based dialogue systems. I summarize the results of these stages next.

### 1.2.1 Literature analysis and critique

The literature analysis and critique presented in Chapter 2 focuses on a representative sample of the large body of work relating to the mechanical generation of cooperative responses. I found that it was possible to divide the literature into three general categories:

1. cooperative question answering for database-query systems;
2. cooperative question answering in task-oriented domains;
3. general theories of cooperative response behavior.

This categorization made it easier to understand the context of individual efforts and to see how that context affected which kinds of responses were considered cooperative.

The critique leads to the conclusion that previous research has been concerned not so much with understanding and modeling cooperative response behavior as it has with designing computer systems that can *manifest* such behavior. That is, the systems are all able to give the appearance of being "cooperative respondents" in the particular query situations for which they were designed, but their behavior does not actually rest on stated principles of cooperation. In fact, for most researchers, theories of cooperation and cooperative question answering are at best only peripheral concerns. Furthermore, the utility of the resulting models has often been dependent on ill-defined properties of particular domains or applications. Because only fixed forms of behavior have been modeled, little attention has been paid to the question of how cooperative behavior is constrained.

### 1.2.2 The questioner-based perspective

Chapter 3 distinguishes two distinct theoretical perspectives on the general CRG problem, each with a unique research methodology. In Section 3.1, I characterize the first of these—the *questioner-based perspective*, a view of CRG through the eyes of the questioner. The term *questioner-based model* is used to describe theories of CRG that view the response-generation task from the questioner's perspective. For questioner-based models, the primary goal is to devise mechanisms that enable dialogue systems to live up to the expectations of the questioner in a given query situation. Principles of cooperation are used primarily as justification for why dialogue systems should behave in the various ways that have been studied and modeled. These principles, however, are hardly ever stated precisely, and they have never played a direct role in determining the behavior of a given system. I argue that, to the extent that researchers have actually proposed accounts of CRG behavior, their theories have all been questioner based.

After characterizing the questioner-based perspective and its associated research methodology, I show that the general questioner-based approach has undesirable theoretical consequences. First, I argue that the results yielded by the questioner-based approach to the CRG problem have turned out to be models of highly specialized *response situations* rather than theories about the causes and motivations of the different examples of cooperative response behavior. Second, I claim that all of the principles of cooperation that have been proposed (or that can be abstracted from previous research reports) to date have been purely *descriptive*, that is, they describe what systems are doing but not why they are doing it.

Taken together, these problems explain the resistance of questioner-based models to integration attempts. Such models show how to provide cooperative responses only in restricted dialogue situations, or in the context of particular question-answering applications. The theoretical results resist integration either because the situations in which they apply are incomparable or because the underlying applications implicitly make different assumptions about the kinds of response behavior their users find helpful. Thus I conclude that the questioner-based approach is of limited usefulness for reaching the long-term engineering goal. Moreover, the questioner-based approach by its very nature ignores the questions whose answers would further the scientific effort.

### 1.2.3 The respondent-based perspective

In Section 3.2, I develop an alternative view on the problem of cooperative question answering: the *respondent-based perspective.* I argue that if we want to be able to build systems that are truly to serve as cooperative respondents rather than as models of particular examples of cooperative responses, we must attempt to model *the respondent's knowledge and reasoning in a dialogue*, instead of her behavior as perceived by the questioner.

The characterization of the respondent-based perspective centers on the principle that *responses reflect the respondent's goals.* This means that the utterances of a natural-language response should be viewed as the observable result of the respondent's execution of a plan she had formed to achieve some set of goals. (These goals, however, may not be immediately evident in the response, but rather constitute high-level nodes in the plan that underlies the actual response.) This idea leads to the dissertation's fundamental hypothesis:

> In order to build dialogue systems that are to be generally capable of acting as cooperative partners in a conversation, we must view their responses as actions planned to achieve one or more goals.

Providing evidence for this hypothesis is a major aim of this research.

In the process of contrasting the two theoretical perspectives, I argue that in all questioner-based models, the goal that the system is attempting to achieve is never defined or formalized. Using Kaplan's COOP system [Kaplan 82] as a case study, I distinguish five elements of a respondent-based model of cooperative response generation:

1. the goals that respondents adopt and try to achieve in their responses;

2. the conditions of the conversation that motivate various response goals;

3. the knowledge and reasoning resources needed to determine the status of the goal-motivating conditions;

4. the reasoning processes and principles by which response goals are actually adopted;

5. the strategies that may be used to achieve a given response goal.

I suggest that a program of research aimed at fleshing out the details of these five theoretical elements will help us both to build more capable cooperative dialogue systems and to better comprehend the nature of cooperative behavior. Section 3.2 concludes with a discussion of several examples that illustrate many complexities of cooperative interaction that can be neither studied nor modeled in a questioner-based framework.

### 1.2.4 Cooperative response planning systems

If responses are to be viewed as attempts to achieve the respondent's goals, it follows that cooperative respondents should be viewed as *planning agents*. This idea is the theme of Chapter 4. In that chapter I outline a general architecture for a *Cooperative Response Planning System* (CRPS), a system whose core component is an agent capable of forming and adopting goals and developing plans to satisfy them.

While the case that Section 3.2 builds in favor of the respondent-based perspective is a strong one, it also raises difficult computational questions. It does not explain how one could actually implement a respondent-based dialogue system. That turns out to be a complex and multi-faceted research problem, and the work in Chapter 4 is thus intended as an attempt to identify the theoretically significant parts of that problem.

The architecture I describe is intended to be a first approximation to a theory of cooperative response planning. The divisions that it imposes on the process of cooperative response planning are derived from observations and analyses of numerous examples of naturally occurring request-response pairs.

At the most general level of detail, I divide the response-planning process into three sequential stages: a process of building a "conversation model" containing beliefs derived from the questioner's utterances (Section 4.2), a process of building a "response plan" comprising a set of goals to be achieved through the response (Section 4.3), and a language-generation process. The chapter concentrates on characterizing conversation models and the process of constructing them, and on the process of constructing response plans.

Although the architecture is not yet sufficiently fleshed out to be used as a blueprint for an actual implementation, it is nevertheless useful in that it seems to expose significant theoretical problems that from a questioner-based perspective would be invisible. Identifying these problems is a useful step toward both the engineering and the scientific goals.

## 1.3 Data Used In Research

Many of the research results presented herein are derived from the study of a wide range of examples of naturally occurring cooperative response behavior. Some of these examples were observed first hand, others were reported to me by friends and colleagues who knew of my interest in the subject, and still others were found in the literature of related work.

But by far my richest source of example data has been two sets of transcripts of naturally occurring dialogue. The first set of transcripts, which will henceforth be referred to as the UNIX transcripts, consists of a body of 103 question-response pairs extracted from a larger corpus of electronic mail messages recorded as they were sent between users and consultants of the UNIX[1] operating system at the University of California, Berkeley. The original transcripts were compiled at UC Berkeley by Lisa Rau.

The UNIX transcripts were compiled in this manner: a central electronic mail address was established ("consult"), to which UNIX users were permitted to send queries. Messages sent to this address were both recorded in a file and distributed to a group of consultants (staff members at the academic computing center). Each recipient could independently

[1] UNIX is a trademark of AT&T.

choose whether to respond. All responses were sent both to the originator of the query and to the consultant distribution list. Thus all consultants were privy to all responses generated to each query. While this usually had the effect of discouraging multiple responses to a given query, in several cases two or more consultants would respond. Those latter few cases, by making it possible to compare responses, have provided an especially rich source of insight into the phenomenon of cooperative dialogue.

The second set of transcripts, hereinafter called the EMACS transcripts, was collected as part of a project for a natural-language seminar during the Spring of 1982 at the University of Pennsylvania [Lewis 82]. Eight dialogues were recorded, each taking place between two people, one acting as an expert and the other as a novice as they communicated via a computer terminal. The user (novice) was assigned a set of text-editing tasks to perform using the EMACS text editor. The user interacted with the expert to obtain information, advice, and assistance on the proper performance of the assigned tasks.

## 1.4 Dissertation Outline

Chapter 2 presents my review and analysis of a representative portion of the literature pertinent to cooperative response generation. Chapter 3 discusses the two opposing perspectives on CRG and argues that a respondent-based approach offers the most hope of research progress in the long term. Chapter 4 examines a proposed architecture design for a cooperative response planning system. Chapter 5 illustrates the architecture's capabilities by describing in detail the processing of two sample responses. The thesis concludes in Chapter 6 with a summary of both some broad conclusions that can be drawn from this work and some specific results, along with discussion of some open questions and interesting directions for further research.

# Chapter 2

# Literature Review and Analysis

One of the many concerns of researchers in natural-language processing has been the development of computational methods to enable natural-language dialogue systems to produce cooperative responses. For these researchers, the term "cooperative response" has usually meant a reply that, for a given question in a particular discourse context, is more helpful or informative than the simple direct answer.

Research in this area over more than a decade has yielded a substantial body of literature. This chapter presents a survey and analysis of a representative portion of that body, the goal being to gauge the amount of progress the field as a whole has made towards a theory of cooperative communication. We will see, however, that any such assessment of progress is surprisingly difficult to make. As the chapter demonstrates, our cumulative understanding of the cooperative response generation process as reflected by the literature is confined to a few informal descriptive principles, a few models of inference, and an assortment of computer programs each simulating a different form of purportedly "cooperative" response behavior, usually in a specific domain or application. The evidence provided in this chapter will three claims:

- Research has focused on modeling *manifestations* of cooperative behavior and rarely considered the nature of that behavior itself.
- These behavior models have often been characterized in ways that are closely tied to a particular domain or application.
- Our comprehension of the *constraints* on cooperative behavior is marginal at best.

The body of literature relevant to cooperative response generation is both extensive and highly diverse. It covers a spectrum of problems arising in different domains of discourse and in the context of different applications. Nevertheless, at least as an organizing principle, the various contributions can generally be divided into three categories: (1) cooperative responses in database-query applications, (2) cooperative responses in task-domain applications, and (3) general theories of cooperative interaction. These are not precise categories, since research efforts do not always fit in a single category (and some do not fit neatly in any category), but they will be helpful at least for distinguishing the general trends of past research. But before proceeding with discussion according to that categorization, I will introduce the Gricean framework in a separate section due to its central role in much of the work on cooperative interaction.

The Maxim of Quantity

Make your contribution as informative as is required.
Do not make your contribution more informative than is required.

The Maxim of Quality

Do not say that which you believe to be false.
Do not say that for which you lack adequate evidence.

The Maxim of Relation

Be relevant.

The Maxim of Manner

Avoid ambiguity and obscurity of expression.
Be brief and orderly.

Figure 2.1: Grice's Maxims

## 2.1 The Gricean Framework

Grice's "cooperative principle" (CP) and its attendant maxims [Grice 75] have had a profound influence on the development of theoretical and computational models of cooperative response generation, perhaps more so than any other theory. The CP is stated as follows [Grice 75, p. 45]:

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

This principle is elaborated in four maxims, named QUANTITY, QUALITY, RELATION, and MANNER, shown in Figure 2.1. Both the CP and the maxims are broadly stated and thus subject to interpretation. At best, they characterize some forms of discourse that arise when agents are acting cooperatively; they do not, however, tell us much about what it means, in a prescriptive sense, to be cooperative. For example, while it is generally true of cooperative interaction that respondents avoid saying things that are false (that is, they appear to obey the maxim of QUALITY), it seems odd to think of that as an axiom of cooperative behavior. Truthfulness should follow from a theory of what it means to be cooperative rather than be a principle partially defining cooperation.[1]

It should also be understood that using the Gricean framework to justify a system's behavior misses the point of Grice's analysis and misunderstands the intended purpose of the CP and maxims. The intent of Grice's seminal article was to reconcile the logic-based approach to formulations of sentence meaning with the wide variety of non-logical

[1] In other words, "I am truthful because I am cooperative", not "I am cooperative because I am truthful."

meanings—*implicatures*, as Grice called them—that sentences can carry. The maxims were actually proposed as "trip wires"—not rules to be followed *per se* but rules that, *when violated* ("flouted," in Grice's term), trigger a very logical inferential process[2] of "working out" the meaning implicated by the violation. Thus rather than being a set of constrained generative principles, the maxims are stated broadly in order to account for as much non-logical inference as possible.

## 2.2 Category 1: Database-Query Applications

Database-query systems have long been a popular application area for natural-language processing technology. They are so attractive largely because (1) the query task is well understood (and in many cases is supported by a well-developed mathematical theory), (2) the database structure provides a complete and precise definition of the domain of discourse, and (3) the majority of the queries can be handled with relatively small lexicons and grammars. One of the notable early successes in this area was LUNAR, a natural-language interface to a database of chemical analyses of lunar rock samples [Woods 72].

Users of database-query systems are treated as simple information seekers: they engage the system in a dialogue solely to obtain information. Any external purpose they might have in obtaining information is just that: external, inaccessible to the system's reasoning, with no influence on either how the question is interpreted or how the answer is found or presented. The user provides to the system a natural-language characterization of the desired *result set*—the elements to be retrieved from the database—and may also propose requirements for how that result set is to be presented. The system then analyzes the user's input and dynamically constructs a procedure that, when executed against the database, should retrieve all and only those elements fitting the user's characterization. That set of retrieved elements constitutes the *direct answer* to the query.

Research in this area has focused primarily on problems in mapping from natural-language structures to formal query languages (e.g., Woods' work [Woods 77] on quantification) and on related issues such as the transportability of interfaces across databases [Grosz 87]. Some investigators, however, have observed that the connection between a natural-language question and an appropriate answer is not always a direct one, and that database interfaces consequently have significant responsibilities in an interaction above and beyond translating the language of the user to the language of the database. This section examines several proposed extensions to the basic interface-as-translator model intended to enable database interfaces to provide extended answers that are not only *correct* but also *useful*, *helpful*, and *informative.*

### 2.2.1 Presumption failure

Kaplan was one of the first researchers to show that it is often insufficient and even inappropriate to give only literally correct answers to natural-language questions [Kaplan 81, Kaplan 82]. He distinguished the *presumptions* of a question from its *presuppositions.* The presuppositions of a question are those propositions that must be true for any direct answer

[2] A process that Grice specified only in very general terms.

to be meaningful. Viewed another way, they are conditions on the felicitous utterance of a question. For example, consider the question shown in (1)a, below [Kaplan 81, p. 128]:[3]

(1) a. Q: What day does John go to his weekly piano lesson?
    b. John takes weekly piano lessons.
    c. R: Tuesday.

The question in (1)a presupposes the truth of (1)b. If (1)b is false, there can be no valid direct answer such as (1)c. This is in contrast with a *presumption*, a proposition whose falsehood entails that *at most one* correct direct answer exists. The proposition in (2)b is presumed but not presupposed by the question in (2)a, for if (2)b is false, (2)c is a literally correct, direct answer (and in fact, it is the *only* answer).

(2) a. Q: Did Donald divorce Marla?
    b. Donald was married to Marla.
    c. R: No.

Kaplan suggested that when people pose questions, they ordinarily do so in the belief that they have left the respondent with a choice of direct answers. Thus questions can implicitly convey to a respondent some of the questioner's beliefs. For instance, a questioner who asks (2)a implies that he believes (2)b. As Kaplan noted, a problem arises when a respondent provides only the literally correct answer (2)c to (2)a. That answer implicitly confirms the proposition (2)b and thereby misleads the questioner. A query having one or more false presumptions is said to exhibit *presumption failure*. Kaplan proposed, as a general rule of cooperative behavior, that when respondents detect presumption failure in a query, they are obliged to respond in a way that at the very least establishes the presumption's falsehood.[4]

Kaplan noticed that it was not uncommon for natural-language queries on databases to exhibit presumption failure. His COOP system was designed to detect and respond appropriately to one typical class of presumption failure in database queries, namely, those that arise when a constituent of a query is found to have an empty result set. Consider query (3)a:

(3) a. Q: Which programmers from the Windows group are in department 67?
    b. R: None.
    c. R: I don't know of any programmers in department 67.

The query literally requests a list of the members in the intersection of two sets: programmers from the Windows group, and programmers in department 67. If either set is empty, the literal answer to the question would be (3)b. But this answer is misleading; (3)c is far more helpful and informative. These failures—which Kaplan called *extensional query failures*—were the kinds of presumption failures that COOP handled.

Although not emphasized in Kaplan's work, COOP was also capable of two other forms of helpful behavior: *suggestive indirect responses*, and *paraphrases*. In certain cases when a

[3]Throughout this dissertation, when referring to the individual participants in a question-answering exchange, I will use the symbol 'Q' to denote the *questioner* and 'R' to denote the *respondent*. When it is necessary to pronominalize them, Q and R will be treated as male and female respectively.

[4]An alternative approach, in which *answers* rather than *questions* are tested for provably false presuppositions, is presented in [Mercer 84].

valid query (i.e., one without a presumption failure) had an empty result set, COOP would try to "generalize" the query by removing a restriction, as illustrated in (4).

(4) Q: Which projects in oceanography does NASA headquarters sponsor?
R: I don't know of any projects in oceanography that NASA headquarters sponsors. But you might be interested in any projects that NASA headquarters sponsors: (list of projects)

Here COOP removed the "in oceanography" restriction on "projects". This process of *query generalization* was later formalized and extended by Motro (see Section 2.2.3).

COOP also routinely paraphrased each query before generating the answer, as shown below.

(5) Q: Who doesn't sponsor projects in the oceanography area?
P: (I am assuming that "oceanography" is an AREA OF INTEREST.) Assuming that there are projects that are in the oceanography area, who doesn't sponsor those projects?

The paraphrase is useful in that it demonstrates both *that* the query was understood and *how* it was interpreted. For ambiguous queries, the paraphrase brings both the nature of the ambiguity and the system's resolution of it to the user's attention. An extended discussion of the COOP paraphraser can be found in [McKeown 79].

### 2.2.2 Intensional query failures

Influenced by Kaplan's work, Mays identified a more general class of presumption failure which he called *intensional query failure* [Mays 80]. According to Mays, a query fails intensionally if it presumes a world model that does not accord with the database schema. For example, the query "Which students teach courses?" presumes that the relation TEACH holds between entities of type STUDENT and entities of type COURSE. But given a database schema in which that relation did not hold, the description "students who teach courses" would be malformed, and thus the literal answer to such a query would be "zero" or "nil". To deal with this, Mays developed a system that could reason from the database model to validate all the relationships that the query presumed. This testing was performed before the query was passed to the database system. If an error was detected, the system generated an appropriate corrective response and aborted the database transaction. This is in contrast with COOP, which was triggered only when a database transaction yielded a null answer.

### 2.2.3 Query generalization

Motro suggested that users of database-query systems usually interpret a null answer (correctly or not) as an indication that their query was improperly constructed [Motro 84]. This has two unfortunate consequences: first, null answers offer the user no assistance in correcting a truly ill-formed query, and second, if the query was in fact well formed, users tend to *misinterpret* the null answer as an error. In light of this, Motro proposed three requirements on how database-query systems should respond when the result set is empty:

such systems should (1) assure the user that the query was meaningful if in fact it was, (2) delimit the scope of a failure, and (3) anticipate possible follow-up queries.

Motro claimed as a general principle that while a questioner might reasonably expect his direct question to have a null answer, he would not expect any *generalization* of that query to have a null answer. One way to generalize a query is to eliminate a restriction (as was done in COOP): the query shown in (6)b is generalized from (6)a in this manner.

(6) a. Q: Which female programmers are assigned to the Frobozz project?
b. Q: Which programmers are assigned to the Frobozz project?

According to Motro's theory, a questioner who asks (6)a might be willing to accept an answer of zero, but would not expect such a reply to the related query (6)b which omits the gender restriction. Consequently, a system should be able to detect misconceptions by testing the generalizations of a query. If a generalization returns a zero or null answer, that generalized query reveals one of the user's misconceptions. So if the user asks (6)a and both it and its generalization (6)b return nil, it can be deduced that the user incorrectly believes that there is a set of programmers assigned to the Frobozz project. This has an important corollary: if all immediate generalizations (since queries can be repeatedly generalized) *succeed*, the original failure was genuine, i.e., the original query was well formed.

Motro developed a query-generalization mechanism and proposed to add it to the database-query processor. Triggered by a null answer, the generalizer can apply various generalization techniques to the original query, such as eliminating restrictions, generalizing along type hierarchies (e.g., generalizing *beer* to *alcoholic beverage* in a query like "Which supermarkets sell beer?"), and relaxing specific numeric values to ranges. If all immediate generalizations succeed, the system can both assure the user that the original query was well formed (in line with Motro's first requirement) and offer the generalizations and their answers as "partial answers" (in line with the third requirement) in the manner of Kaplan's suggestive indirect response, e.g., "No supermarkets sell beer, but the following five sell alcohol", followed by a list. If any generalization fails, that new query is recursively generalized until a non-failing query is reached. The system meets Motro's second requirement by using the most general failing query to form a misconception correction.

### 2.2.4 Exploiting integrity constraints

In some database systems, domain-specific constraints are defined and used both to maintain database consistency across updates and to improve query optimization. Gal observed that these *integrity constraints* can be exploited to answer questions cooperatively and informatively [Gal 85, Gal 88]. The particular application she studied was a question-answering system on a university course-registration database.

Suppose that the system has a constraint to the effect that Professor Jones teaches only in the Ancient History department. Ordinarily a query like (7)a would have to be answered as in (7)b, but that is not very helpful. Instead, Gal suggested that the known semantic constraint be used to produce a more cooperative response such as (7)c.

(7) a. Q: How many Middle East Studies courses does Jones teach?
b. R: None.
c. R: None, because Jones only teaches Ancient History courses.

Unlike the systems of Kaplan, Mays, and Motro discussed earlier, Gal's use of integrity constraints is potentially helpful whether the query succeeds or fails, as in example (8).

(8) Q: How many courses does Jones teach?
R: He teaches only in Ancient History; this semester, he teaches 4 courses.

Here the query has succeeded (the system was able to determine the number of courses taught by Jones), but the system was able to incorporate the known semantic constraint to produce a more informative response.

Gal's system is capable of several other forms of cooperative response behavior as well. One of these involves pragmatically interpreting yes/no questions as wh-questions when the answer is affirmative, as shown in (9).

(9) Q: Do some history courses have an enrollment of more than 50 students?
R: Yes, courses 120 and 220.

This behavior is an aspect of what was previously called *over-answering* yes/no questions in [Wahlster 83]. In another form of cooperative behavior (also described previously as *providing additional case-role fillers* in [Wahlster 83]), the focus of wh-questions is enlarged to incorporate other variables, as in (10).

(10) Q: Which courses does Mudbrick teach?
R: SC301 in Digital Design, and SC550 in Software Engineering.

The principle here is that the query implicitly requests the department as well as the course number. Lastly, the system is able to replace lists by more general descriptive responses (a desirable feature also noted in [Finin 86]):

(11) Q: Which graduate courses are taught by professors?
R: All graduate courses are taught by professors.

By exploiting the semantic constraint, the system not only provides a concise reply, but also avoids misleading the user: a list of all the graduate courses might imply to the user that one or more graduate courses exist that are *not* taught by professors.

### 2.2.5 Monitor offers on dynamic databases

Mays's doctoral thesis [Mays 84a] investigated cooperative question answering on a *dynamic database*: a database that both records its history of modification and incorporates a model of possible future change.[5] In particular, Mays noted that question-answering systems built on dynamic databases could provide *monitor offers*: offers to provide information in the future when it becomes available (see also [Mays 82]). Consider Example (12):

(12) Q: Is the U.S.S. Scorpion within 50 miles of Portsmouth?
R: No, but would you like me to let you know when it is?

[5] See also [Clifford 88], which presents a model of question answering (although without considering cooperativeness) for *historical relational databases*, databases that record their history but, unlike dynamic databases, do not model their possible futures.

The system answers the question and additionally offers to monitor the database and notify the user when the event of interest occurs.

Mays observed that only certain kinds of queries could merit monitor offers in response. The query in (13), while it has the same structure as the one in (12), illustrates a case in which a monitor offer would be inappropriate:

(13) Q: Is San Francisco within 50 miles of Portsmouth?

Mays suggested that monitor offers could be classified as either *competent* or *incompetent*, and developed a modal temporal logic for axiomatizing database change, thereby making it possible to compute whether or not a potential monitor offer was competent. A monitor offer was judged incompetent when it could be proved (from the temporal axiomatization and the current state of the database) that the expected event would never occur.

### 2.2.6 Anticipating follow-up questions

While not strictly a database application like those we have already seen, the HAM-ANS system of Wahlster *et al.* nevertheless dealt with issues that are equally pertinent to those systems [Wahlster 83]. HAM-ANS was a natural-language interface to a vision system, permitting users to ask questions about scenes and visual events processed by the vision component.

The particular cooperative behavior examined here was called *over-answering* yes/no questions by anticipating possible follow-up questions. Two forms of this behavior were identified: (1) responding with a more specific quantifier, and (2) providing additional case-role fillers. Both of these were later used by Gal, and having discussed them earlier, I will leave it to the interested reader to seek out Wahlster *et al.*'s original exposition.[6] The same authors did, however, recognize the potentially unconstrained nature of over-answering, and proposed an *informativeness-simplicity trade-off* as a bounding principle. The trade-off was specified in three heuristics:[7]

1. Avoid superfluous complexity, i.e., complexity not justified by any increase in informativeness.

2. Do not allow a certain maximum degree of complexity to be exceeded.

3. Within these limits, maximize the amount of information presented.

The main difficulty with these heuristics is that there does not seem to be any principled methodology for implementing them in a given domain. The second heuristic, for example, implies the use of an arbitrary fixed threshold value. There is a question as to whether the choice of this value is domain dependent. Indeed, it is probably more realistic for such a value to be computed dynamically; how that might be done for a given query in a given domain, however, is an open question. While these heuristics *describe* how the response behavior of HAM-ANS is constrained, it is not at all obvious how they might be used or how helpful they might be for other systems.

[6] An extensive application of over-answering is also discussed in [Eugenio 87].

[7] These heuristics were characterized as "possible instantiations of the Gricean Maxim of Quantity" [Wahlster 83, p. 645].

### 2.2.7 Answering questions about database structure

In its final form, McKeown's TEXT system concerned matters of cooperative behavior only peripherally; first and foremost it was a contribution to the study of natural-language generation [McKeown 85]. However, the motivation for TEXT came from the desire to find ways to extend the usefulness of natural-language interfaces to databases.[8]

McKeown suggested that, at least in complex database applications, users should not be expected to know the database schema. Rather, interface systems should permit users to ask questions not only about the database's *contents*, but also about its *structure*. The TEXT system was able to answer three kinds of questions:

- requests for the *definition* of a database concept C;
- requests for a *description* of the information available about a concept C;
- requests for a *comparison* of two database concepts.

These query types correspond to queries like, "What's a submarine?", "What do you know about submarines?", and "What's the difference between a missile and a bomb?"

### 2.2.8 Describing the result set: levels of abstraction

In all the database systems that have been described so far, the notion of what constitutes an *individual* was taken as a given. For example, in a university course-registration database such as Gal's, the individuals are the courses and the professors. Database queries then request different sets of individuals.

Corella has demonstrated that the concept of an individual is not always so clear [Corella 85]. He studied a database system that tracked inventories of TTL semiconductor devices. In this database, each TTL device entry consisted of a serial number, a version (standard, high power, low power, schottky, or low-power schottky), a grade (military or commercial), and an ordering code. Working in this framework, Corella evaluated possible responses to the request, "List the flip-flops."

This request can be answered in various ways, for example, as a list of serial numbers (14), as a list of codes combining serial number and version (15), or perhaps as a list showing serial number, grade and version (16).

(14) R: 70, 71, 72, 73, ...
(15) R: 70, H71, 72, H72, 73, H73, LS73, ...
(16) R: 5470, 7470, 54H71, 5472, 7472, 54H72, 74H72, 5473, 7473, 54H73, 74H73, 54LS73, 74LS73, ...

In other words, the notion of an individual is no longer fixed. Rather, it varies according to what Corella called the relevant *level of abstraction*. The idea is that, given a query, there is a particular level of abstraction at which the system should make its reply. Corella was

[8] McKeown's work began as a study of cooperation in database-query applications. Her early studies of transcripts collected by Malhotra [Malhotra 75] showed that users frequently requested both definitions of terms and comparison of terms. TEXT was originally an attempt to build that capability into a database-query system. But as often happens in research, the final product made its main contribution in an entirely different area, in this case, natural-language generation (Bonnie Webber, personal communication).

able to provide a formal definition for the level of abstraction of an answer, but he suggested that determining from a query the appropriate level of abstraction for the response is a very difficult task indeed. This remains an open problem.

### 2.2.9 Summary and comments

As we have seen, research on cooperative question answering in database-query applications has focused mainly on the development of techniques to make system responses more useful, helpful, and informative, with several efforts aimed at preventing the generation of misleading answers. The following principles either were proposed in or can be abstracted from the work discussed in this section:

1. Correct false presumptions (intensional, extensional) to avoid misleading the user.
2. Offer partial answers when the direct answer is zero or null.
3. Delimit the scope of query failures.
4. Anticipate follow-up questions.
5. Respond at the appropriate level of detail.

As encouraging as these results might seem, it is not clear how they could be applied outside the confines of database query to more general models of discourse. Stenton's comment in this regard is apropos [Stenton 86, p. 103]:

> The aim of [research on cooperative question answering] is to improve the communication capabilities of expert and database systems in order to facilitate the problem solving tasks for which they are used. To achieve this within AI, attempts have been made to exploit the conventions of human dialogue. These have not always been principled. That is, it is unusual for the underlying principles of co-operative communication to be fully specified in the analysis. A common approach seems to comprise: the identification of certain dialogue structures as "co-operative"; and the subsequent development of programming techniques which generate them.

Indeed, it is often difficult to see how the different efforts relate to one another. Some of the contributions seem to depend upon unique assumptions about the underlying database (e.g., Mays's dynamic database, Gal's integrity constraints). To what larger purpose, if any, do these results then contribute?

Simply put, because the domain of database querying is so restricted, researchers are able to engineer "helpful" and "cooperative" response behavior. This domain does not, however, force researchers to probe beneath the surface into the causes and motivations of that behavior. It also does not encourage researchers to investigate what limits might need to be placed on their models of cooperative behavior: any needed limits are provided implicitly by the application.

## 2.3 Category 2: Task-Domain Applications

One of the most important conceptual entities missing from database-query applications is the user's *goal*. As was pointed out earlier, database-query systems never consider *why* a given query is being asked. But the true nature of cooperation seems fundamentally to relate to how agents help each other achieve goals. Database-query applications equate the user's goal with knowing the correct answer to the question (and perhaps also to not being misled). Systems that provide extra "helpful" information (as in Gal's system and HAM-ANS) are able to do so only because their designers have exploited *a priori* knowledge of the typical relationship between queries and users' higher goals. This knowledge, however, is never represented or reasoned from explicitly in these systems.

In *task-domain applications*, the execution of some underlying task is central to the purpose of the interaction. The most common experimental task-domain application is the *expert advisor*. These systems are typically intended to provide "expert" advice to people engaged in tasks of various sorts, like assembling a complex mechanical device, filling out a tax return, or planning a university course schedule. Unlike database-query systems, expert-advisor systems must explicitly represent and reason about goals and users' progress towards them in order to help users complete their tasks. Task-domain applications that demand natural-language interaction are fruitful domains for research into the nature of and constraints upon cooperative behavior in general and cooperative response generation in particular.

Various task-domain applications have been described in the literature. They have differed from one another in terms of the characteristics of the underlying task (e.g., physical tasks versus mental tasks), the discourse roles of the system and user (expert/apprentice, advisor/client, etc.), the time at which the task is to be executed (co-temporal with the dialogue versus after it), and the relative amounts of knowledge held by system and user. We will find that the work reviewed in this section (with the exception of Kidd's research discussed in Section 2.3.5) has mostly focused on identifying and modeling new aspects of cooperative behavior that are desirable in task-domain applications.

### 2.3.1 Correcting object-related misconceptions

In virtue of the more complex domains in which they typically operate, expert-advisor applications engage in discourses that are considerably more elaborate than those engaged in by database-query applications. As a result, the variety of misconceptions that can possibly be detected and corrected during a conversation is correspondingly richer. One new type of misconception, called an *object-related misconception*, has been identified by McCoy [McCoy 85, McCoy 88]. As she observed, dialogue participants sometimes explicitly or implicitly reveal that they have a misconception about a conceptual object (as opposed to a specific instance of an object) in the domain. Two kinds of object-related misconceptions were distinguished: *misclassifications* and *misattributions.*

McCoy characterized these misconception types in terms of a taxonomic domain model defining the domain concepts, their attributes and subsumption relationships. According to this model, a misclassification misconception exists when a user incorrectly believes that some entity $E$ is of type $C$, when in fact $E$ is actually of type $C'$ such that the classes of objects $C$ and $C'$ are disjoint. For example, a user might misclassify a whale as a fish when the domain model records it as a mammal. A misattribution occurs when a user incorrectly ascribes a property to an object that according to the domain model does not have that property. So a user might err by attributing the property "has gills" to whales.[9]

McCoy's ROMPER system was conceived of as a module in a larger dialogue system, a module called upon to generate a corrective response should some other (unspecified) process detect an object-related misconception of either of the two kinds studied. The core of the system is a set of correction schemas to be instantiated in the context of the misconception. Besides the schemas themselves, McCoy's principal contribution in this area is the development of a technique called *object perspective*, used to highlight the context-relevant concepts and attributes of the domain model to be used when instantiating the appropriate schema.

McCoy's work was inspired by consideration of some of the novel demands that will be placed upon discourse systems operating in task domains. The work focuses, however, on only a single manifestation of cooperative response behavior: the correction of a particular kind of misconception. No claims are made about when object-related misconceptions occur, how they might be detected, when a correction is required, or how a corrective response might be integrated with other response goals that a larger dialogue system (within which ROMPER is supposed to be but one of many modules) might be trying to achieve.

### 2.3.2 Correcting plan-related misconceptions

Quilici *et al.* describe an explanation-based approach for detecting and responding to misconceptions about plan *applicability conditions*, *enablements*, and *effects* [Quilici 88]. The authors' claim is that "a cooperative response is one that not only corrects the user's mistaken belief, but also addresses the missing or mistaken user beliefs that led to it" (p. 38). Their AQUA system is conceived of as an expert advisor to help users solve problems they have encountered while using the UNIX operating system. Users describe to the system a problem they have encountered while trying to achieve some goal. A typical example is shown in (17).

(17) Q: I tried to remove a file with the "rm" command. But the file was not removed and the error message was permission denied. I checked and I own the file. What's wrong?

R: To remove a file, you need to be able to write into the directory containing it. You do not need to own the file.

From the description,[10] AQUA infers a set of beliefs that account for why the user thought the given file-deletion command he used was an appropriate means for achieving the goal. The set of inferred beliefs is then used to identify the underlying abstract cause of the problem, permitting the system to produce a corrective response such as the one in (17).

Their approach is to define a small set of predicates for encoding the plan-related beliefs of an advice seeker (shown in Figure 2.2) and then to combine those predicates to form

[9] This is a case in which the misattribution follows from a deeper misclassification: once the user misclassifies whales as fish, he proceeds to wrongly attribute piscine properties to them.

[10] The AQUA system does not actually deal with natural-language input. Instead, the relevant beliefs are hand coded.

| | |
|---|---|
| `causes(A,S)` | Executing A has effect S. |
| `!causes(A,S)` | Executing A does not have effect S. |
| `enables(S1,A,S2)` | S1 is necessary for A to have S2 as an effect. |
| `!enables(S1,A,S2)` | S1 is unnecessary for A to have S2 as an effect. |
| `applies(A,S)` | A is a correct or normal plan for achieving state S. |
| `!applies(A,S)` | A is not a plan for achieving S. |
| `precludes(S1,S2)` | S1 and S2 cannot hold simultaneously. |
| `!precludes(S1,S2)` | S1 and S2 can hold simultaneously. |
| `goal(A,S)` | Actor A wants to achieve S. |

Figure 2.2: Representation of planning relationships in AQUA

*explanation patterns*—schematic configurations of user and system beliefs that represent different general categories of plan-related misconceptions. These patterns can be applied both to identify the misconception (by matching the abstract configurations against the inferred explanatory beliefs) and to highlight the (correct) advisor beliefs that should be presented to the user as a corrective, "cooperative" response.

The product of Quilici *et al.*'s research is a classification of some general mistakes that users might make in forming their plans. But "being cooperative" is equated here solely with correcting misconceptions (as their claim implies). The detection by AQUA of a plan-related misconception is considered sufficient reason for it to instantiate and generate a corresponding response pattern. There is no reasoning about how the existence of the misconception relates to the questioner's goals or how his goals might be furthered by the correction; the system's response to a detected misconception is "cooperative" only in the eyes of an outside observer.

### 2.3.3 Intelligent agents as the basis for dialogue systems

Chin has argued in his dissertation that in order to take the initiative in dialogue, to be able to do such things as correct misconceptions and volunteer information instead of answering queries passively, a dialogue system must be constructed as an *intelligent agent* with its own goals and plans [Chin 88a]. His UCEGO system was designed to provide a capacity for intelligent agency in UC (the UNIX Consultant), an expert advisor for tasks in the UNIX domain [Wilensky 88].

UCEGO makes three main contributions: (1) it identifies several different ways in which cooperative response goals are formed and adopted during response processing, (2) it suggests some of the ways in which the responding agent's own goals might interact with the goals ascribed to the questioner and thereby influence the choice of response, and (3) it shows how a multi-variable user model (called KNOME) can be used to help the system identify misconceptions and make text-planning decisions.

The UCEGO program itself can be thought of as a *blackboard system*[11] [Nii 86b]. An encoding of the input (in the KODIAK semantic-network formalism [Wilensky 87]), along

[11] This analysis of UCEGO as a blackboard system is entirely my own.

with the inferred plan and goal of the user, is placed in "working memory" (i.e., on the blackboard). Various knowledge sources, which Chin terms *if-detected daemons*, are then triggered by patterns of the system's beliefs stored there. All the system's behavior is implemented by the operation and interaction of if-detected daemons.

Chin divided if-detected daemons into five categories: *themes*, *foreground* and *background goals*, *sub-goals*, and *meta-goals*. *Themes* are very general goals such as "act like a consultant" and "be polite." By default, the "consultant role theme" is in effect when UCEGO is invoked, giving rise to a *foreground* goal of greeting the user and offering assistance. While foreground goals are planned for directly, *background* goals are planned for only when demanded by conversation-specific conditions. For example, UCEGO has a general "self-preservation" theme; under certain triggering conditions, this theme causes one or more background goals to reach foreground-goal status. That is, UCEGO never directly plans to achieve self-preservation goals, but instead holds self-preservation as a background goal. When its existence is threatened (e.g., the user asks, "how do I crash UCEGO?"), self preservation becomes a foreground goal that the system then works to achieve.

*Sub-goals* in UCEGO are what one would expect them to be: goals to be achieved in the process of achieving a more general goal. *Meta-goals* are the means by which UCEGO deals with goal *conflicts*. For example, when the user asks, "how do I crash UCEGO?", a conflict arises between the system's goal of answering the question (which follows from its consultant-role theme) and its goal of self preservation. When such conflicts occur, the system adopts meta-goals to resolve them. In the self-preservation case, UCEGO decides to abandon its goal of answering the user's question and adopt instead a goal of *refusing to answer*. Once the system has selected its goals, a planner is invoked to achieve them by means of natural-language expressions.

Examples (18)–(21) demonstrate some of UCEGO's response abilities.

(18) Q: How can I delete a file?
R: Typing "rm filename" will delete the file filename from the current directory.

(19) Q: What does ls -v do?
R: There is no -v option to ls.

(20) Q: Does ls -r recursively list subdirectories?
R: No, it reverses the order of the sort of the directory listing; ls -R recursively lists subdirectories.

(21) Q: How can I delete my supervisor's files?
R: You shouldn't be deleting other people's files.

The approach embodied in UCEGO is a step in the right direction. There can be no question that cooperative dialogue systems will have to model both their own goals and the goals of their users in order to achieve generality. It is important to understand how these goals interact and how that interaction leads to cooperative behavior. Unfortunately, Chin does not develop any principled theoretical basis for defining if-detected daemons. Although the daemons appear to be declaratively specified (in the KODIAK representation formalism [Wilensky 87]), the only motivation for their design seems to be that they make the chosen set of example dialogues work.

As a particular illustration, UCEGO is claimed to have an ability to act "ethically." This cashes out operationally as follows: suppose the user asks how he might crash the computer. The user's goal of crashing the system gets posted on UCEGO's blackboard. UCEGO has an "ethics" theme, encoded in an if-detected daemon, that gives rise to goals of preventing users from damaging the computer system. When the user asks to be informed how to crash the system, UCEGO adopts the goal of preventing him from acquiring that information. This ultimately leads the program to issue a refusal to answer the question.

But is the system really acting ethically, or is it only giving the *appearance* of ethical behavior? I would argue that the latter is the case. The "ethics" daemon is a domain-dependent condition-action rule that makes the system act in the desired way but otherwise lacks generality and hence cannot be transferred straightforwardly to other domains. The point is that UCEGO has no idea *why* it should prevent users from achieving **crash-system** goals; rather, an if-detected daemon checks for the KODIAK equivalent of "user wants to crash the system," and, when such a pattern is found, yields the goal "UCEGO prevent user from crashing the system." But a general theory of ethical behavior must at least be grounded in an understanding of the possible consequences of various actions. The other rules (daemons) in the system suffer from a similar lack of generality.

The explanation for the weakness is simple: the research focus is on different issues. UCEGO is designed to exploit a single computational mechanism: the if-detected daemon. All of the system's functionality is implemented by rules encoded as daemons. The research investigates not the principles that might be used to identify the daemons that are needed to support cooperative interaction, but rather the kinds of interactions that might occur between daemons (via their modifications to the blackboard). Thus no particular effort was made to motivate or justify the actual daemons used, beyond the fact that they made the system behave as desired.

### 2.3.4 Knowledge-intensive planning

Luria, also working within the UNIX Consultant framework, has described the design of a *commonsense planner* and outlined its role in cooperative question answering [Luria 88]. According to Luria, commonsense planners differ from other kinds of planners in that they are *knowledge intensive*, having access to a great deal of domain-specific information. The main problem in knowledge-intensive planning is determining the information that is relevant to the problem-solving task at hand. His program, called KIP, has three tasks within UC:

1. to select the set of user goals for which a plan is to be developed;
2. to synthesize a plan that the user can carry out to achieve those goals; and
3. to check the resulting plan for potential execution failures.

KIP's initial goal or goals are derived from the user's utterance. It is assumed that these goals are achievable, and that the user simply lacks a plan for them. Goals for KIP are relatively simple sorts of things that one might want to do in a computing environment, like spool a file to a printer, edit a file, or delete a file. Plans are sequences of operating-system commands.

The research focuses on how the planner builds plans and, in particular, on how during its operation the planner can identify sources of potential execution failure and cause them to be brought to the user's attention. KIP is able to distinguish between *actual* and *potential* sources of failure. As an example of the former, the planner might notice while constructing a plan for printing a file on a given printer that the printer is out of paper (the planner must have access to knowledge about the dynamic state of the printer in order to detect this). Here, KIP would not only present the plan but also would advise the user to first refill the printer's paper tray. As an example of the latter, the **mv** command, used to rename (move) files, is known to have a possibly undesirable side effect: if one moves a file *file1* to a file *file2*, and *file2* exists at the time of the move, it will be lost and replaced by *file1*. KIP is aware of this, and so when presenting **mv** plans to the user, it is able to provide appropriate warnings.

KIP operates by instantiating and manipulating data structures called *concerns*. Concerns encode the system designer's domain-specific knowledge about likely sources of plan failure. During both plan synthesis and checking, KIP searches its library of concerns for any that are relevant to the current plan. When found, a concern may cause KIP to adopt an additional goal for which to construct a plan. Example (22) exhibits a typical response that KIP is able to produce.

(22) Q: How do I print Jim's file on the lineprinter?
R: Print Jim's file foo by typing lpr foo. This plan will not work if you don't have read permission on Jim's file or if the printer is out of paper.

The issue relevant to the purposes of this chapter is the nature of KIP's contribution to our comprehension of cooperative interaction. That, I think, can be summarized as follows: in task-oriented domains, respondents are often called upon to synthesize and present plans of action to the questioner (most often when responding to "how" questions). During this process, they might notice ways in which the plan they intend to propose either *might* fail (if certain conditions hold) or *will* fail (unless certain conditions are eliminated). The (implicit) claim is that under such circumstances, the respondent is obliged to bring the relevant conditions to the user's attention.

### 2.3.5 Deriving principles experimentally

Kidd is noteworthy in this area for her effort (the only one of which I am aware) to verify experimentally that people do indeed consistently generate different classes of cooperative response in different situations [Kidd 90]. Her experiments were conducted in the computer-configuration domain; ten test subjects played the role of expert in configuring a particular model of computer. In order for them to be actual experts (or at least to make a good show of it), they were provided with all necessary guidebooks and reference materials for configuring a particular computer and given some time to familiarize themselves with the task.

The test subjects were then run through a series of configuration problems. They were presented with various initial configurations and in those contexts were asked questions taken from three question categories (shown in Figure 2.3). The subjects were told that these questions came from salespeople, but in fact they were designed by Kidd. The experiments yielded interesting patterns of cooperative response behavior. For example, consider the Category 1 question shown in (23). Test subjects were asked this question in a context where the proposed action was not possible. Kidd found that the simple negative

1. Verify that some action is possible (e.g., "can I do X?")
2. Given some proposed action, verify whether some other action is necessary (e.g., "if I do X, do I have to do Y?")
3. Elicit the number of times a proposed action is possible or necessary (e.g., "how many X?")

Figure 2.3: Question Categories in Kidd's Experiment

answer was often (73% of the time) augmented to include information about the condition preventing the action, as it was in this case:

(23) Q: Can I add 80Mb of memory?
R: No, you are already at the limit of added memory.

The experiment generally showed a strong correlation between the initial state of the planning space and the ways people chose to modify or augment their responses. Her results, Kidd argued, suggest that it should be possible to formulate a predictive theory of cooperative response for this task domain. She also claimed her results show that for such well-defined reasoning tasks as computer configuration, systems can generate cooperative responses without having to dynamically recognize the user's goal or maintain an extensive dialogue history.

Once more it should be clear that we are dealing here only with manifestations of cooperative behavior. When Kidd speaks of a "predictive theory", she means a set of condition-action rules that can be used to engineer "cooperative responses". These rules are to be derived from experimental observations. But the question is: will these rules tell us anything about cooperative response behavior, or will they merely help us build systems that appear to be cooperative? If the latter, it becomes a matter of concern as to how well the rules will carry over to other domains.

It is significant that the computer-configuration problem considered by Kidd is about as well structured as any task-domain problem could be. The plan library (of configuration rules) is both complete and finite, and there is only one goal: to reach a valid configuration. Both advisor and user share complete knowledge at all times of the state of the configuration: there is no possibility for a belief discrepancy or misconception. The user's knowledge of the configuration rules is assumed to be a proper subset of the advisor's, and it is the missing knowledge that motivates the dialogue. Questions in Category 1 are asked because the user does not know whether all the preconditions of the proposed action are satisfied; questions in Category 2 concern the effects and ramifications of particular actions; questions in Category 3 inquire into general domain-imposed limits. Given these restrictions, it should not be too surprising that specific cooperative principles can be stated. In the general case, however, the user is likely to have other goals that impose constraints on the configuration process. For example, the user might want to minimize cost, maximize processing capacity, or evaluate different trade-offs. Once we allow some of the natural complexities to enter the picture, it seems less likely that response rules will be simple functions of the question and the initial configuration, or that the resulting rules can be transferred straightforwardly to other domains. Nevertheless, the work is promising and it would be helpful to see more studies like it.

### 2.3.6 Summary and comments

We have evaluated several research projects conducted in task domains. The first two systems reviewed—ROMPER and AQUA—considered how some new kinds of misconceptions that arise in these domains might be detected and corrected. UCEGO explored the proposal that cooperative response systems be designed as goal-driven agents. KIP showed how various forms of domain-specific knowledge about plans and actions could be encoded and used to increase the informativeness of answers to "how" questions. Finally, Kidd analyzed a particular task domain experimentally and identified regularities in the kinds of cooperative responses given to various question types under different conditions.

It should be evident from the discussion that all these efforts have modeled cooperative response behavior from the questioner's point of view. This is to say that they have identified particular forms or ranges of forms of response behavior desirable to the questioner, then designed the computations needed to generate them. The goal was to reproduce the behavior, not to specify the reasoning that might have motivated it. As a result, it was never important or necessary to state the principles from which the behavior followed; it was sufficient to characterize the "cooperativeness" of the responses in terms of how they might affect or be perceived by the questioner. (I will elaborate on this point in Chapter 3.)

The next section focuses on projects that have attempted to probe more directly the nature of cooperative interaction, either by studying the properties of general reasoning mechanisms and formalisms or by developing theories that account for some range of examples of cooperative behavior.

## 2.4 Category 3: Theoretical Approaches

All research efforts reviewed in this section, while many are framed in terms of examples drawn from particular task domains, are more correctly regarded as attempts to develop domain-independent models of different aspects of cooperative communication. I begin with a discussion of several projects that have developed plan-based theories of cooperative interaction. A few less easily categorized projects are discussed in the remainder of the section.

### 2.4.1 Models based on planning and plan recognition

It has long been an accepted hypothesis that cooperative interaction between agents is intimately connected to their ability to recognize and accommodate each other's goals and plans. Some of the first steps toward formalizing this principle in a manner sufficient for use in a computational setting were taken by Allen and Perrault [Allen 78, Allen 80], Cohen [Cohen 78], Cohen and Perrault [Cohen 79], Sidner and Israel [Sidner 81], and Sidner [Sidner 83]. The first two of these, Allen and Perrault, developed a theory in which speech acts [Searle 71] were used as the formal basis for inference of the speaker's intentions.

(This work later developed into Allen's dissertation, to be discussed shortly.) Cohen and Perrault [Cohen 79] outlined a formalization of speech acts in a planning framework (to be discussed next). Sidner and Israel introduced the notion of the *intended meaning* of an utterance in a discourse, and considered how recognition of that meaning might influence the response. I begin with a discussion of Cohen and Perrault's plan-based model of speech acts, then review several efforts that have fleshed out the role of plan recognition in cooperative question answering.

### A plan-based model of speech acts

Austin claimed that speakers use their utterances to perform actions, such as requesting, promising, and asserting [Austin 62]. Searle elaborated Austin's theory and attempted to specify sets of necessary and sufficient conditions on the felicitous performance of several general categories of speech acts, including *directives* (e.g., request, command), *commissives* (promise, threaten), *declaratives* (utterances like "I now pronounce you man and wife" that have the effect of changing the state of the world), *representatives* (inform, deny), and *expressives* (thank, apologize) [Searle 69]. Cohen and Perrault further formalized these ideas by developing the first *competence theory* of speech acts—a theory modeling the possible intentions that underlie speech acts [Cohen 79].

In general, human problem-solvers can be viewed as agents who form and execute plans. These plans specify action sequences that transform an initial world state to a desired goal state. The approach that Cohen and Perrault have taken is to model speech acts as actions in a planning system. One advantage of this approach is that it enables the planning of both linguistic and nonlinguistic acts to be treated uniformly.

Formal models of planning usually represent actions as *operators* that are defined in terms of three elements:

**Preconditions** The *preconditions* of an action are those states of the world that must be true for that action to be executable.

**Effects** The *effects* of an action are those states of the world that are made true as a result of the successful execution of the action.

**Body** The *body* of an action describes, at an increased level of detail, the method by which the action's effects are achieved. Actions that an agent can execute directly, without further decomposition into substeps, are said to be *atomic*.

Cohen and Perrault hypothesize that people maintain symbolic descriptions of the world and that those models contain descriptions of the world models of other people. They view speech act operators as primarily affecting the models that speakers and hearers maintain of each other. They limit their formalization of a plan-based theory of speech acts to REQUEST acts, INFORM acts, and questions. They claim that these acts (a) appear to be definable solely in terms of speakers' and hearers' beliefs and goals, (b) have a wide range of syntactic realizations, and (c) are thought to comprise a significant proportion of ordinary discourse. My review will focus only on their formalization of REQUEST and INFORM acts, for in that work can be found all the theoretical elements that are relevant to the present concerns.

The authors distinguish between *competence* and *process* theories of speech act planning. Competence theories, in their view, attempt to characterize the structure of well-formed plans that might underlie speech acts, whereas process theories focus on the steps speakers might follow when choosing their speech acts and the mechanisms that hearers use to recognize those acts. Cohen and Perrault claim to have developed an initial competence theory of speech act planning.

The models that agents maintain of each others' beliefs are represented (following [Hintikka 69]) using the modal operator BELIEVE, where "A BELIEVE(P)" means that agent A believes the proposition P. Goals are represented using the WANT predicate: "A BELIEVE B WANT P" represents agent A's belief that agent B has goal P.[12]

Operator schemata in the proposed planning theory record the operator's effects and distinguish two kinds of preconditions: "cando" and "want" preconditions. CANDO.PRs schematize propositions that, when instantiated with the values bound to the parameters of the operator instance, yield conditions that must be satisfied in the world model for that operator instance to be executable. WANT.PRs are used to characterize the intentional nature of actions; they formalize the conditions that define the planning agent's intention to perform the action. Note that both CANDO.PRs and WANT.PRs can be recursively planned, i.e., a planning agent can extend her plan with actions whose effects correspond to the CANDO.PRs and WANT.PRs of other actions.

In elaborating Austin's theory, Searle defined different sets of necessary and sufficient conditions on the successful performance of speech acts. These conditions fall into the following categories:

**Normal I/O Conditions** These conditions aim to ensure that the communication situation is "normal," e.g., the speaker can speak and the hearer can hear, the speaker and hearer are not actors in a play, and so forth.

**Propositional Content Conditions** Only certain kinds of propositions are appropriate for each type of speech act. For example, the propositions communicated by commissive acts must predicate some obligation of the speaker.

**Preparatory Conditions** Preparatory conditions define the world states that must hold for the speech act to be felicitously performed.

**Sincerity Conditions** Sincerity conditions distinguish sincere performances of speech acts from insincere performances.

**Essential Condition** The essential condition specifies what the speaker is trying to accomplish by performing the speech act.

**Force Condition** The force condition (Cohen and Perrault's term) is used ensure that speech acts are uttered only if the speaker intends to communicate that he is performing that act.

In Cohen and Perrault's formalization, propositional content conditions are implicit in the schematization of speech act operators. Preparatory conditions are mapped to the preconditions and essential conditions to the effects of speech act operators. Normal input/output

[12]The authors note that the formal semantics of WANT are problematic.

conditions are assumed, and the force condition is ignored, since that depends on a better understanding of the process of speech act recognition.

| REQUEST(S,H,ACT) | |
|---|---|
| CANDO.PR: | S BELIEVE H CANDO ACT |
| | AND |
| | S BELIEVE H BELIEVE H CANDO ACT |
| WANT.PR: | S BELIEVE S WANT request-instance |
| EFFECT: | H BELIEVE S BELIEVE S WANT ACT |

| INFORM(S,H,PROP) | |
|---|---|
| CANDO.PR: | S BELIEVE PROP |
| WANT.PR: | S BELIEVE S WANT inform-instance |
| EFFECT: | H BELIEVE S BELIEVE PROP |

Figure 2.4: Initial definitions of REQUEST and INFORM acts

The authors' initial definitions of the speech acts operators for REQUEST and INFORM are shown in Figure 2.4. The parameter S denotes the *speaker*, H the *hearer*. Note that as defined the effects of the two speech act operators are not the ones for which such acts would typically be selected. A speaker makes a request not only to cause a hearer to believe that the speaker wants some act to be performed, but also to cause the hearer to decide to perform that act. Similarly, inform acts are performed to cause hearers to believe the asserted propositions. To mediate between the speech act definitions in 2.4 and such desired effects—called *perlocutionary effects*—Cohen and Perrault define the intermediate step CAUSE-TO-WANT for REQUESTs and CONVINCE for INFORMs. These two steps are shown in Figure 2.5.

| CAUSE-TO-WANT(A,B,ACT) | |
|---|---|
| CANDO.PR: | B BELIEVE A BELIEVE A WANT ACT |
| EFFECT: | B BELIEVE B WANT ACT |

| CONVINCE(A,B,PROP) | |
|---|---|
| CANDO.PR: | B BELIEVE A BELIEVE PROP |
| EFFECT: | B BELIEVE PROP |

Figure 2.5: Mediating between REQUEST/INFORM and perlocutionary effects

The definitions shown in Figures 2.4 and 2.5 are the authors' first formalization of

Searle's theory of speech acts. They proceed to evaluate these definitions with respect to various adequacy criteria and discover that some revisions are necessary. First, they compare the planning of a REQUEST to the planning of an INFORM of a WANT, pointing out that "please do $x$" and "I want to you do $x$" can have the same force as directives. They find that in the latter case, their formulation allows the speaker to overlook her beliefs about the hearer's beliefs about his ability to perform the act. This motivates a redefinition of both REQUEST and its mediating CAUSE-TO-WANT step.

Second, they explore the consequences of composing REQUESTs to form third-party requests such as "Ask Tom to tell me where the key is". They discover that when such an utterance is issued to an intermediary, say John, their formalism places unnecessary requirements on John's beliefs. In particular, their formalism demands that John believe that Tom (the target agent) knows the location of the key. This condition should not be required since John is performing the REQUEST act on the speaker's behest; all John need believe is that the *speaker* believes that Tom knows where the key is.

This problem is overcome by a combination of two tactics. First, Cohen and Perrault rightly note that the performance of speech acts causes hearers to acquire certain beliefs about the speaker's beliefs. That is, the performance of a speech act may have *side effects.* For example, one side effect of a REQUEST act is that the hearer H comes to believe that the speaker believes that H CANDO the specified act.

As their second tactic, the authors adopt a neutral "point of view" for defining the CANDO.PRs and EFFECTs of speech act operators. They call this the *point-of-view principle*:

> The "Cando" preconditions and effects of speech acts should be defined in a way that does not depend on who the speaker of that speech act is. That is, no CANDO.PR or EFFECT should be stated as a proposition beginning with "SPEAKER BELIEVE." (p. 206)

The revised definitions of REQUEST, CAUSE-TO-WANT, and INFORM are shown in Figure 2.6.

**Summary** Cohen and Perrault were the first to formally model the goals that hearers (as respondents) adopt in reaction to speakers' utterances. By formalizing speech acts in a planning framework, they provide a tidy and illuminating model of the beliefs and goals that speakers must have for them to felicitously choose to perform request and inform acts or to issue certain *wh* questions. It should be clear, however, that the issues they address are largely orthogonal to those which are of concern to developers of cooperative natural-language dialogue systems. It is true, for example, that their formulation of REQUEST precludes speakers from requesting a hearer to perform an act that the speaker believes the hearer cannot do (or that the speaker believes the hearer believes he cannot do). But this is a theory of utterances' *felicity* rather than their *cooperativeness.*

Central to any theory of cooperative interaction is the question of how a speaker arrives at her *intentions* to perform the kinds of speech acts that Cohen and Perrault model. In their plan-based formalization, REQUEST and INFORM acts have certain WANT.PRs. Theories of cooperative response behavior need to elucidate the principles from which cooperative conversational partners come to have the goals that WANT.PRs express, that is, the theories must explain why and how speakers decide to WANT to perform REQUEST

| REQUEST(S,H,ACT) | |
|---|---|
| CANDO.PR: | H CANDO ACT |
| WANT.PR: | S BELIEVE S WANT request-instance |
| EFFECT: | H BELIEVE S BELIEVE S WANT ACT |

| CAUSE-TO-WANT(A,B,ACT) | |
|---|---|
| CANDO.PR: | B BELIEVE A BELIEVE A WANT ACT<br>AND<br>B BELIEVE B CANDO ACT |
| EFFECT: | B BELIEVE B WANT ACT |

| INFORM(S,H,PROP) | |
|---|---|
| CANDO.PR: | PROP |
| WANT.PR: | S BELIEVE S WANT inform-instance |
| EFFECT: | H BELIEVE S BELIEVE PROP |

Figure 2.6: Revised definitions of REQUEST, CAUSE-TO-WANT, and INFORM

and INFORM acts. Cohen and Perrault's work should therefore be understood as taking over where theories of cooperative reasoning leave off, as defining the linkage between intentions to communicate and speech act utterances.

### Plan-based reasoning for obstacle detection

Allen's model [Allen 83] has been as influential on the development of plan-based theories of cooperative behavior as Grice's maxims have been on theories of cooperative interaction in general. Allen proposed a theory of plan recognition intended to explain (1) how people choose to provide more information in a response than the questioner explicitly requested, (2) how people interpret sentence fragments, and (3) how people understand *indirect speech acts* [Searle 75]. It is how the model relates to the first item that is of interest here.

Allen defined an *obstacle* as a situation preventing an actor from achieving a goal, and claimed that "many instances of helpful behavior arise because the observing agent recognizes an obstacle in the other agent's plan, and acts to remove the obstacle" [Allen 83, p. 108]. One class of obstacles was considered: those related to an agent's lack of knowledge. For example, not knowing the telephone number constitutes an obstacle to calling someone by telephone. Two kinds of knowledge obstacles were identified: *explicit* and *implicit.* Explicit obstacles are indicated by the user's question, since it identifies knowledge that the user lacks and that he needs in order to successfully execute his plan. Implicit obstacles are detectable only by inspecting the agent's underlying plan of action; e.g., from a request for a train's time of departure, a respondent might recognize that the questioner is planning to board the train and from that deduce that he needs to know both the train's departure time and its departure gate.

The core of Allen's recognition system was a set of domain-independent plan inference and plan construction rules. A heuristically-guided algorithm searched for an inference chain connecting the user's utterance to one of a set of domain-specific *expected goals.* The prototype system played the role of an information-booth attendant in a train station, a domain which admitted two possible expected goals: boarding a train and meeting a train.

At first glance it would appear that Allen concentrated on three very specific forms of language behavior, but in fact the work should be regarded as a general theory on account of its domain-independent formulation of plan recognition as an information provider for cooperative response generation. The main limitation of this theory is one shared by all that have come after: the basic task performed by plan recognition systems is searching, and no one has found principles for controlling the search in anything but small domains with representations of limited expressive power.[13] This remains a challenging open problem.

The general cooperative principle proposed here, namely, that cooperative agents detect and act to eliminate obstacles detected in other agents' plans, has substantial intuitive appeal. It accounts for many examples of both linguistic and non-linguistic cooperation.[14] It is worth noting, however, that it assumes that the "other agent" has a *valid plan* (see the discussion of Pollack's work in the following section). If the other agent's plan is not useful (for instance, it will not achieve the desired goal), it surely makes no sense to attempt to remove obstacles from it, if indeed the concept of an "obstacle" is well defined in such a situation.

It is also worth noting that the principle offers little guidance as to how much effort a respondent is required to expend in her search for obstacles. Plans are, at least in principle, infinitely decomposable. How thoroughly should an inferring agent check an actor's plan for obstacles? This question will most likely be answered when legitimate search control techniques are found. I conjecture that the answer will at least partly depend upon an understanding of the extent to which the respondent is *responsible* for acting on behalf of the questioner. For example, a train information clerk need not consider a passenger's (potential) lack of a ticket to be an obstacle, as both participants understand that she is not responsible for providing one.[15]

### Recognizing ill-formed plans: plans as mental phenomena

In her dissertation, Pollack criticized earlier models of plan recognition (such as Allen's) that made what she dubbed the *valid-plan assumption*—the assumption that the actor's plan is valid [Pollack 86]. She argued that this assumption must be abandoned if natural-language systems are to achieve human-level competence in cooperative dialogue. Consider the paradigm example shown in (24)–(25).

[13] Indeed, the standard "search control technique" is not really a technique at all: one merely defines a limited knowledge base and allows the inference mechanism to search it exhaustively.

[14] Here is a typical example of non-linguistic cooperation: one agent, observing another agent approach a door with her arms full, opens the door for her to pass through.

[15] She may, however, need to treat as an obstacle the passenger's lack of knowledge about how or where to obtain a ticket.

(24)a. Q: I want to prevent Ira from reading my personal mail files.
b. How do I change the access codes to private?

(25)a. R: Give the command 'chmod go–rwx file.'
b. Unfortunately, you cannot stop Ira from reading your files. Because he's the system manager, he can override file permissions.

In (24)a, the questioner establishes his immediate domain goal: denying Ira access to Q's mail files. The query in (24)b is well formed since plans exist in the domain for changing file-access codes to private; R could, for example, synthesize a plan for that goal and supply it to Q, saying something like (25)a. But the query must be interpreted in light of the stated goal rather than in isolation.[16] Plan recognition here serves to relate the two utterances via an underlying plan in which Q intends to achieve the deny-access goal by achieving the change-permission goal.

The significant point here is that the plan underlying the question is *ill formed.* Were R unable to recognize this situation (because she reasoned using a plan recognition formalism that made the valid-plan assumption), she would be able to answer the question only as in (25)a, without giving the more informative response in (25)b.

Pollack's solution starts from the view that plans are mental phenomena, configurations of beliefs and intentions that an inferring agent ascribes to an actor. By viewing plans this way, plan recognition models can treat ill-formed plans as plans in which discrepancies exist between the beliefs of the inferring agent and those that she ascribes to the actor. In other words, plan inference becomes a process of belief ascription (an idea that [Konolige 89] explores further).[17]

Pollack's particular approach was to formalize the theory of action *generation* proposed by Goldman [Goldman 70]. Informally, an action $\alpha$ is said to generate another action $\beta$ if by performing $\alpha$ an agent performs $\beta$ as well. For example, the act of flipping a light switch can, under the right conditions, generate the act of turning on a light.

Pollack focused on *simple plans*—plans in which all actions are related by generation—and defined what it means for a simple plan to be *ill formed* as opposed to *well formed*, what it means for an action in her framework to be *unexecutable*, and what it means for a query to be *incoherent.* Given a stated domain goal (as in (24)a) and a request to be informed how to do a particular act (as in (24)b), Pollack's SPIRIT system infers the simple plan that relates them, then identifies any discrepancies that exist between the beliefs of the system and those ascribed in the simple plan to the user.[18] The system presents all relevant information in its response: (1) if the queried act is doable, the user is informed how to do it, otherwise the fact of its unexecutability is asserted; (2) each invalidity found in the plan is noted; (3) if the plan was ill formed but SPIRIT knows of another way to achieve the goal, it presents that information.

[16]It is usually argued that this claim follows from general principles of rationality. That is, believing the questioner to be a "rational agent", we expect his utterances to cohere. We therefore prefer interpretations that find a relationship between utterance pairs such as (24)a–b over interpretations that treat them as unrelated.

[17]In general, this approach raises difficult theoretical problems, most having to do with how the inferring agent comes by her knowledge of the actor's beliefs, and, in particular, how the inferring agent makes well-motivated decisions about whether a belief discrepancy exists.

[18]Some of these discrepancies were detected with the assistance of a *user model.* See [Rich 79] and [Kass 88] for some thorough treatments of user modeling issues.

SPIRIT and its corresponding theory were motivated by the general idea that a respondent's awareness of the invalidities in a questioner's plan influences her choice of cooperative response. The work contributes to our understanding of how cooperative respondents come to notice those invalidities, i.e., how they acquire the knowledge they need to meet their cooperative responsibilities. Pollack makes this explicit [Pollack 86, p. 76–77]:

> I take the intuitively perceived regularities in, and differences between, responses found in naturally occurring discourse to be evidence for structural regularities in, and differences between, the plans that are inferred to underlie queries. In other words, I am willing to make assertions of the following form: if a response to some query exhibits a certain feature (say $F$), one can account for that response by claiming that the plan inferred to underlie the query is invalid in some particular way (say, it has an invalidity of type $I$). However, I will not make converse assertions: I do not claim that whenever the plan inferred to underlie a query is invalid in some particular way ($I$), a cooperative response to it must exhibit a particular feature ($F$).

Thus Pollack's efforts add to our understanding of the reasoning mechanisms that support cooperative behavior, but they tell us very little about the principles that *drive* that behavior.[19]

### Adapting plan-recognition theories to discourse

Carberry, noting that models of plan recognition such as Allen's and Pollack's are limited to isolated utterances, has been adapting these theories to extended discourse [Carberry 83, Carberry 88, Carberry 90]. She has concentrated on task-oriented domains in which users ask questions that are related to plans to be executed after the discourse concludes (she calls these *information-seeking dialogues*). Her examples are largely in the domain of course-advising for university students. For example, an undergraduate might ask questions about particular courses, their prerequisites, instructors, and so on, in order to find out how to obtain a bachelor's degree.

The central insight of this work is that plan recognition during a discourse must be performed incrementally. No longer can it be assumed that a complete plan is inferred from a single utterance. Instead, each utterance provides clues only to a portion of that plan. The problem, then, is discovering how to integrate the information derived from successive utterances into an evolving model of the user's goals and plans (which Carberry terms the *context model*).

Carberry's inference system builds context models using a two-step approach. First, after each input utterance, Allen-style inference rules [Allen 83] are applied to hypothesize a set of *candidate focused goals* (*cfg*'s) and corresponding *candidate focused plans* (*cfp*'s), the domain-dependent goals and plans that each utterance indicates the user's attention may be focused on. *Focusing heuristics* are then used to decide where in the context model to attach a given *cfp*. The derivation of multiple *cfp*'s from a single utterance signals that several underlying plans could account for the utterance. In this situation, the context

[19] For a completely different take on the problem of recognizing ill-formed plans, one based on probabilistic reasoning, see [Calistri 90].

model is copied for each *cfp*, and the attachment procedure is run separately for each pair of context model and *cfp*.[20]

Because the user's plan is to be executed at some later time, his line of questioning is not tightly bound to some underlying task-execution order (as in the kind of expert-apprentice dialogues studied by Grosz and Sidner [Grosz 77, Grosz 81, Grosz 86]). Instead, the user can shift his attention among various substeps in a larger plan. Happily, there *do* appear to be constraints on possible shifts, and that is where the focusing heuristics come into play. They reflect these constraints and provide guidelines that help the system to relate each new utterance to the context established by previous utterances.

There is, however, an interesting flaw in Carberry's model. Recall the purpose that people have in engaging in the kind of dialogue at issue here. As Carberry says [Carberry 88, p. 2]:

> We are interested in a class of information-seeking dialogues in which the information-seeker is *attempting to construct a plan* [emphasis mine] for a task that will be executed after the dialogue terminates.

A user who is attempting to *construct* a plan for a given goal (such as obtaining a degree at a university) by definition does not have that plan fully formed in his mind. Thus his questions may be more likely to suggest parts of plans *under consideration* than parts of a plan the user might actually be intending to execute. For example, an inquiry into when a particular course meets might be less of an indication of an intention to take that course than an indication that the meeting time of the course will affect whether the user will decide to take that course (for example, the user might sensibly prefer afternoon courses to morning courses). In order to cope with such discourses, a plan recognition model would seem to have to distinguish the language used to talk *about* the user's task plan from that task plan itself. In other words, it would need a language in which plans and their components were first-class objects.[21]

But Carberry's system has no such meta-language. Lacking the means to reason about different elaborations of the underlying plan, the model ends up assuming—being *forced* to assume—that the user enters the discourse with a plan that is complete (at least structurally) rather than under construction. Consequently, the model in fact only allows queries about parameter bindings at nodes in this plan. This problem is reflected in the focusing heuristics: consider the example in (26)a–c (assume that (26)a–b constitute a single "turn", and that (26)c is the questioner's subsequent question):

(26) a. I want to major in computer science.
b. When does CS120 meet?
c. How about CS180?

After (26)b, the system will infer that the user intends to take CS120 as part of a plan to satisfy a major in computer science. The focusing heuristics (specifically, Carberry's

[20] Carberry's model has little to say about how this ambiguity might be resolved in general. In her system, ambiguous context models are only eliminated when a subsequent utterance causes the system to detect an inconsistency. She states that in designing her system this way she has made the valid-plan assumption (see Section 2.4.1, page 36), though she later discusses how this assumption might be relaxed in her framework. More recently, Eller has proposed a ***meta-rule*** approach to error recovery [Eller 90].

[21] Litman has taken steps toward the development of such a language [Litman 84b], but with an entirely different motivation.

Heuristic 2 [Carberry 88, pp. 7–8]) will cause the succeeding utterance (26)c to be attached in the context model as a *sibling* of the "take CS120" goal. There is no way in the model to treat "taking CS120" and "taking CS180" as *alternatives under consideration.* What if both courses met at the same time? In principle, the plan inferred by the system would be inconsistent. The detection of such an inconsistent plan could very well induce the system to make an inappropriate reply along the lines of "You can't take both CS120 and CS180." None of Carberry's later proposals for how the system might recover from errors introduced into the model avoid this problem: it is an inherent property of the system.

Of course, adding a meta-language would complicate the theory. It is not obvious how the system might be able to distinguish intended plans from plans under consideration.[22] It is also not clear how important such a distinction would be. The only way I see to determine that would be to apply Carberry's theory to some naturally occurring discourses and check at each turn whether the contents of the system's context model squared reasonably well with the observed response.

It also seems interesting that the model does not account for the role of the *system* in the discourse. It would seem that, at the very least, the answers the system supplies to the user's queries should influence its expectations (encoded in the focusing heuristics) of future utterances. In this regard, Litman's discourse model may be helpful [Litman 84b].

Finally, we turn again to considering the contribution this work makes to the theory of cooperative behavior. Clearly, for advisory systems in task domains, the ability to generate helpful, "cooperative" responses is essential. Carberry has cogently made the point that such systems realistically must be thought of as engaging in discourse rather than as answering isolated questions. From this view it follows that plan recognition, widely believed essential to both discourse understanding and cooperative response generation, must be adapted to the discourse model of interaction. But it is plain from its omission of the system as a discourse participant that Carberry's model, like all the other work on plan recognition, makes no commitments as to where plan recognition fits in the larger model of system-as-respondent, as to how the derived knowledge influences the choice of a cooperative response, or as to what "being cooperative" is really all about.

### Coping with ambiguity

An interesting contribution to theories of both plan recognition and cooperative response generation has recently been made by van Beek and Cohen [vanBeek 90]. The authors claim that a questioner's utterances may not always be consistent with only a single plan. For example, they argue that a questioner who asks, "Can I drop Numerical Analysis?", may be trying to avoid failing, trying to eliminate a scheduling conflict, or trying to switch to a more interesting course. Other models of plan recognition have tended instead to return only single plans, a condition often enforced by heuristic means (e.g., Allen's system [Allen 83]), by using measures of likelihood (e.g., Calistri's probability-based PATHFINDER system [Calistri 90]), or even by implicit design (e.g., Pollack's SPIRIT system ends its search when the first complete plan is found [Pollack 86]).

Van Beek and Cohen argue that it may be possible to generate cooperative responses

[22] The theory of meta-plans developed by Ramshaw [Ramshaw 89], which he applied to the task of interpreting ill-formed utterances, may be helpful in this regard. His more recent work on modeling plan *exploration* [Ramshaw 91] looks especially promising.

without necessarily having to determine the exact plan the questioner intends to execute. To this end, they provide a procedure to decide whether resolution of the ambiguity matters to the response.

Their model presumes a plan-recognition system based on the theory developed by Kautz [Kautz 87]. Given a query, the system infers a set of possible underlying plans. Next, the plans in this set are *critiqued*, that is, examined for faults and annotated accordingly if any are found. The decision procedure is then applied to determine whether the ambiguity must be resolved before the system may try to compute a response. Should the ambiguity need to be resolved, the authors propose that a clarification subdialogue be entered (and they provide an algorithm for the clarification procedure).

As far as theories of cooperative response generation are concerned, van Beek and Cohen can be seen as claiming that cooperative response generation systems need to be able (1) to recognize cases in which the questioner's utterances are consistent with several plans, (2) to determine whether any ambiguity affects the system's choice of a response, and (3) if necessary, engage the questioner in a clarification subdialogue to resolve the ambiguity.

The paper has a number of theoretical weaknesses (which is neither unreasonable nor unexpected in a workshop submission). For example, although the authors suggest that the process of resolving plan ambiguity partly depends on reasoning about the query, their definition of the decision procedure makes no reference to it. The paper also fails to maintain a careful distinction between *intended* plan recognition and *keyhole* recognition [Cohen 81].[23] It is generally accepted that intended recognition is the appropriate model for cooperative interaction, since questioners try to make as much of their plans and goals recognizable to a respondent as they believe the respondent needs to provide a helpful answer. Kautz' theory, however, is a keyhole recognition model. Thus the authors' decision to base a theory of cooperative question answering on Kautz' system is a questionable one.

Despite the weaknesses in the underlying theory, the claims of the paper are moving in the right direction. However, I would frame the problem rather differently. I believe that we ought to maintain the principle that questioners aim to make their pertinent goals and plans recognizable *unambiguously* to a respondent, and thus that the plan-inference model appropriate to cooperative question-answering situations is intended recognition. Consequently, it is eminently reasonable for the plan-inference component in a question-answering system to return a single plan—exactly the plan that the respondent believes the questioner intended her to recognize. In fact, this is actually a desirable behavior, since some queries (e.g., "Do you have change for a dollar?") are potentially consistent with a huge number of plans. Given this view, the problems that the authors attribute to plan ambiguity should be modeled differently.

Let's consider the authors' example, "Can I drop Numerical Analysis?" If this query were uttered in a neutral context, a respondent could do no more than infer that the questioner would like to drop the course and wants to know if it is possible (for example, to verify that the deadline for dropping courses has not passed). The respondent's problem is deciding whether there are any reasonable higher goals that the questioner could be

[23]Briefly, intended recognition describes cases in which one agent makes a deliberate attempt to convey his plans and goals to another. Keyhole recognition describes situations in which one agent infers the goals and plans of another agent by observation, where the observed agent is unaware of the observation and thus makes no attempt to make his plans and goals apparent.

trying to achieve (such as not failing the course) which might be thwarted or simply not achieved by the inferred plan. Computationally, this is equivalent to the respondent's making controlled extensions to the inferred plan, rather than generating and testing all possibilities as the authors propose. In other words, I am suggesting that the problem be viewed not as one involving "plan ambiguity" but as a situation in which a respondent actively considers, based on her more complete knowledge of the domain, the possible undesired consequences of the questioner's inferred plan.

#### General comments on plan recognition

This section has considered several different models of plan recognition. They differed partly according to whether plan recognition was to be used as a theory of *understanding* or as a knowledge-acquisition process for *responding*.

Allen's work is interesting in that it relates plan recognition to both understanding and responding. He takes "understanding" an utterance to mean being able to construct a representation of a plan reflecting the underlying intentions. Reasoning from that representation, the system should be able to respond helpfully. Allen focused on a particular aspect of helpful response behavior based on the notion of obstacle elimination (although obstacles were never defined more formally than as unbound variables found along the direct inference chain connecting the utterance to the expected domain goal).

More so than the others, Pollack seems to have been primarily concerned with developing a plan inference formalism to meet the putative needs of the cooperative response generation task. She recognized that the choice of a cooperative response is often influenced by the inferring agent's beliefs about errors present in the actor's domain plan. But her work suggests only how an agent might come to acquire some of those beliefs, and explicitly avoids detailed consideration of how those beliefs might influence the formulation of a response. Instead, SPIRIT says everything that could possibly be relevant.

Carberry focused entirely on plan recognition as a theory of discourse understanding. In her system, the evolving context model represents the system's understanding of the discourse. The question of how that model affects the system's choice of response is not considered.

Van Beek and Cohen raised some interesting points about how cooperative response generation systems might reason about plans but, as we saw, some theoretical difficulties need to be resolved before their proposal can be thoroughly evaluated.

My conclusion about the work on plan recognition is that while all the efforts provide insight into the kinds of knowledge about an actor's plan that might be inferred and represented by a system, none offer good general principles guiding how that information should be used to help a system decide how to respond.

### 2.4.2 Preventing false inferences

As I have been arguing, little progress has been made toward identifying the principles that drive cooperative response behavior. The plan-recognition efforts are unique in their attempt to characterize an information-bearing reasoning process that is intended to support a wide range of behavior, even if those efforts have in practice been restricted to carefully chosen (or constructed) examples in tiny domains.

Besides Allen's suggestion that cooperative agents act to eliminate obstacles detected in agents' plans, only one other principle of cooperative behavior stands out in importance. This principle has shown up so often in the literature that it has become a *de facto* axiom of cooperative interaction. Although its influence can be seen in earlier efforts, Joshi was the first to propose it explicitly. Claiming that successful communication depends in part on the maintenance of consistent *mutual beliefs* between the discourse participants [Joshi 82], Joshi suggested that when one discourse participant comes to believe that there is or might be a discrepancy in mutual beliefs—a discrepancy that might interfere with successful communication—that participant should take the initiative to "square away" (that is, ensure agreement on) the relevant beliefs. This process of "squaring away" mutual beliefs was claimed to account for certain instances of clarification subdialogues.

Joshi noticed that some of the necessary response behavior supporting mutual-belief maintenance was not covered by Grice's theory. Rather than abandoning that theory, Joshi instead proposed an accommodating revision to Grice's maxim of QUALITY [Joshi 82]:

> 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 that inference.

Without this revision, the act of "squaring away" mutual beliefs would constitute an apparent violation of the maxim of QUANTITY, as it leads to additional information not motivated by any other maxim. Joshi argued that the revision to QUALITY was a more principled solution to this problem than some kind of *ad hoc* alteration of QUANTITY.

Joshi's revision, a directive that question-answering systems act to "prevent false inferences", justifies *ex post facto* the behavior of numerous systems. For example, the corrective responses generated by Kaplan's system COOP (Section 2.2.1) block false inferences concerning the non-emptiness of entity sets implicit in the query. Mays's system (Section 2.2.2) blocks false inferences about relationships between database entities, and Gal's system (Section 2.2.4) uses integrity constraints to similar ends. All the work on misconception correction in task domains can be motivated in terms of this principle. Indeed, Pollack's model of plan inference (Section 2.4.1, page 36) supports response behavior that is motivated by Joshi's principle.

Joshi, Webber, and Weischedel [Joshi 84a, Joshi 84b, Joshi 87] have characterized several reasoning tasks that question-answering systems may need to perform in order to keep their replies from being misconstrued.[24] These include blocking false conclusions that might arise from misapplication of default rules (also discussed in [Stenton 86]) and blocking false conclusions in task environments based on expectations about how an expert would respond. In considering these behaviors they rightly note [Joshi 84b, p. 134]:

> The problem is that a system neither can nor should explore all conclusions a user might possibly draw: its reasoning must be constrained in some systematic and well-motivated way.

Unfortunately, neither they nor anyone else has been able to make progress on this problem. This is primarily because research remains restricted to domains that are insufficiently rich for any such constraints to be meaningful or useful.

[24] See also [vanBeek 87] for a discussion of some extensions to this work to accommodate the user's more general goals, plans, and preferences.

### 2.4.3 Search spaces

Allport and Kidd have recently argued [Allport 89] that cooperative response behavior in planning domains can be usefully characterized as reasoning about search spaces. They take a "planning domain" to be a domain in which the basic task is to reason about precondition-action-postcondition sequences and answer queries about the relevance of various actions to a user's goals. Examples drawn from the domain of computer configuration (see also Section 2.3.5) are used to bolster their claims. In this domain, users ask about the possibility, usefulness, and consequences of actions (adding/removing components to/from a computer configuration) with respect to the goal of specifying (for purchase) a properly-configured computer system. Sample query/response pairs are shown in (27)–(29).

(27)a. Q: If I add 4 HP-FL cards, do I have to configure CIB adaptor 0-2?
  b. A: No, you can place one in CIB 0-1 and the remaining three in CIB 1-1.
(28)a. Q: Can I add 2 HP-IB cards in CIB 0-1?
  b. A: Yes, but the system performance will be better if you add them in CIB 1-1 which is less full.
(29)a. Q: If I add 1 device on the HP-IB card in CIB 0-1, do I have to specify option 393?
  b. A: Only if that device is a page printer.

They analyze the reasoning underlying each question-response pair as a search problem. The initial state of the world is modeled as a node in the search space. The query is then a problem statement, mapping to "some number (zero, one or many)" (p. 1) of other nodes in the search space (solution states). In addition, domain or task-specific knowledge may permit other nodes to be designated "of interest to the user", relevant somehow to the user's stated problem. In other words, the query together with any available background knowledge yields a set of nodes in the search space (goal worlds) of interest to the user. There will then be zero or more paths connecting the initial node to the goal nodes. About this Allport and Kidd make the following claim (p. 1):

> A concise but informative description of this set of paths (i.e., this part of the search space) is precisely what constitutes a co-operative response [in examples (27)–(29) above].

It is understandable why this theory of search spaces should be a reasonable way to think about question answering in the domain of computer configuration: Allport and Kidd have found that it is possible to model computer configuration in terms of a STRIPS-style [Fikes 71] state-space planning paradigm. Each world state consists of a description of the user's partially-configured computer (one could easily use a first-order language for this). In the domain there is a set of actions (adding/removing a component) each of which can be modeled as a function between world descriptions (configurations). There is only one goal (a valid configuration), and it can be modeled as a set of propositions that must hold in a world state. Queries become tests on the set of proofs of the goal, starting either from the current configuration as in (28) or from a proposed configuration derived from the current one as in (27) and (29).

While the search-space theory may provide an interesting conceptualization of the computer-configuration problem, it does not seem as helpful as the authors would like to claim for understanding the basic problems of cooperative response generation. For example, while plan recognition can undoubtedly be viewed as a process of finding the relevant paths between an initial state and a goal state (or possibly a set of goal states), does that help us understand its role in the formation of a response, or give us a better idea of the kinds of information plan-recognition systems should yield? In other words, how does the search-space view pay off? My conclusion is that more problems are swept under the rug here than are addressed. Certainly the theory does not bring us closer to an understanding of the nature of cooperative behavior; it only relabels the problems.

## 2.5 Concluding Remarks

As the reader will recall, the declared purpose of this chapter was to defend, based on a literature review and critique, these claims:

- Research has focused on modeling *manifestations* of cooperative behavior and rarely considered the nature of that behavior itself.
- These behavior models have often been characterized in ways that are closely tied to a particular domain or application.
- Our comprehension of the *constraints* on cooperative behavior is marginal at best.

I return to these claims now and argue that they are in fact reasonable conclusions to draw from the preceding discussion.

### Focus on manifestations

When I say that researchers modeled "manifestations" of cooperative behavior, I mean that they designed systems that were able to behave, from the user's point of view, in a cooperative, natural manner. But the systems did not actually reason about what "being cooperative" in the given situation involved. The investigator had already identified the distinguishing features of a query context (e.g., a direct answer of "zero" or "nil") that, in the interests of cooperativeness, motivated some particular modification to or augmentation of the system's default response. The system merely implemented the feature detection and corresponding response alteration.

We saw this approach in the database-query applications. In each case (except in McKeown's and Corella's work), the query coupled with the state of the database comprised a context that could be tested for various features. When the feature of interest was found, the response was appropriately modified (e.g., a monitor offer was provided).

The situation was the same for the task-domain applications. Both ROMPER and AQUA were designed to enable systems to appear cooperative, but neither's behavior followed from cooperative principles. UCEGO did at least make some interesting claims about how intelligent interfaces should reason, but it was rather casual in providing a principled basis for that reasoning. KIP focused on cooperative aspects of responses to "how" questions, but when it modified its responses to warn about potential or actual plan failures, it did so only because the program was designed to perform the required checks, not because it had reasoned that such warnings would be cooperative. Lastly, Kidd proposed that cooperative response behavior be engineered based on observations of human question-answering behavior, an approach that pointedly ignores the issue of why at an abstract level cooperative respondents choose to reply in the ways they do.

To the extent that the more theoretically-oriented efforts led to actual response-producing systems, we saw that they too gave only the outward appearance of being cooperative.

At the risk of beating a dead horse, I will conclude this part of the argument with a quote from [Clifford 88, p. 16–17]:

> Approaching [the issue of the semantics of questions] from an entirely different perspective, researchers in artificial intelligence (AI) have over the years developed and implemented automatic question-answering theories and systems to varying degrees of success. These have ranged from some early experimental programs [Green 63] to database querying programs bound to a particular database domain [Woods 72, Waltz 78] to some rather sophisticated [database query] systems today that are designed to be general and easily portable [Harris 73, Hendrix 78]. The research behind these systems seems to share a goal common to much of the work in AI (as distinct from cognitive science), i.e., an interest more in getting a system to "work" than in developing a formal theory that explains its behavior.

It is precisely because in most systems a rational theoretical basis for their behavior is conspicuously absent that I contend they at best model only manifestations of cooperative interaction.

### Domain-dependent characterizations

For obvious reasons, only the application-oriented efforts in the first two categories have tended to depend on properties of the domain or of the application itself.

For example, in the database-query domain, we saw that Gal's database-query system relied on domain-specific knowledge in the form of integrity constraints. Mays developed a general-purpose modal temporal logic that was to be used to encode a domain-dependent axiomatization of database change, from which it would be possible to reason about the competence of monitor offers. HAM-ANS's filling of optional case-slots in verb case-frames was particularly domain dependent. If, for example, the user asked "Has a yellow car gone by?", HAM-ANS would respond (assuming a yellow car had in fact gone by) "Yes, one yellow one on Hartungstreet." In trying to verify whether an instance of the particular type of locomotion had occurred in the analyzed image sequence, HAM-ANS necessarily had to identify the spatial location of the event, i.e., "the LOCATIVE slot in the case frame for 'to go by' is filled as a *side effect* of the search for an answer" [Wahlster 83, p. 643]. This information was then added to the response. But this "cooperative" act is not based on any reasoning about the questioner's goals, and so there is no guarantee that the extra information will be of any actual interest to the questioner.[25] In truth, the act is only cooperative because users who ask such questions tend also to be interested in knowing the location of the event, as opposed to, say, the car's license number or velocity. But this is domain-specific knowledge that is not encoded anywhere in the system.

[25] Noted also by Chin [Chin 88b, p. 258].

In the task-domain applications, both McCoy and Quilici claimed to have developed domain-independent models of misconception correction. McCoy's claim in this regard is more credible than Quilici's, however. The representation used in AQUA (see Figure 2.2) is presented without any motivation beyond the fact that it works and expresses relationships useful to the theory. The predicate semantics are informally defined, and it is not clear that the predicates actually provide a reasonable way to encode the requisite domain knowledge (certainly no principled encoding method is ever described).

Neither UCEGO nor KIP make any claims to domain independence. Although the general ideas in both efforts (goal-driven agency for UCEGO and KIP's notion of concerns) are not particularly tied to their UNIX application domain, the theoretical details as embodied by the respective implementations are unquestionably so. In UCEGO's case, the domain-specific knowledge is encoded in the if-detected daemons, the rules defining all the system's behavior. In KIP, it is the concerns that are explicitly domain dependent. In neither system are there any significant domain-independent parts.

### Lack of constraints

The point about lack of constraints is simply that past efforts never had to consider how a system's cooperative behavior might need to be limited. We have a sense that in general discourse there is a highly context-dependent line that, when crossed, causes an otherwise cooperative response to seem uncooperative, or at the least undesirable. This intuition is captured in Grice's Maxim of QUANTITY (Section 2.1). Yet in none of the systems we have reviewed in this chapter was there ever a practical possibility that this boundary might be exceeded; thus there was never a need to consider how a system might recognize it mechanically and alter its response behavior accordingly. As discussed earlier, Wahlster *et al.* at least faced the issue (see Section 2.2.6), but their solution appears to engender more questions than it puts to rest.

# Chapter 3

# Two Perspectives on Cooperative Response Generation

In the previous chapter, I examined a wide range of research projects all of which were to some extent concerned with modeling cooperative communication. These efforts primarily described techniques that were developed to improve the "cooperativeness" of responses from natural-language dialogue systems. The review demonstrated, however, that "being cooperative" rarely meant more than simulating a single form or restricted range of forms of response behavior. Moreover, what made these forms of behavior "cooperative" was a matter more often of intuition than of definition: the general principles underlying a given behavior were usually left unstated.

Why do we study cooperative response generation (CRG)? To their credit, other researchers have made useful contributions by classifying, cataloging, and developing computational models of the myriad ways in which respondents accommodate questioners' needs, goals, preferences, etc., in their responses. But if, as I believe, our ultimate goal is to build natural-language systems that are to be more generally capable of acting as cooperative conversation partners, the entire research problem must be approached from a new direction.

In this chapter, I will characterize two opposing perspectives on cooperative response generation, distinguished by the point of view—questioner's or respondent's—from which principles of cooperative reasoning are defined. I will argue that *questioner-based* models define cooperation in terms of kinds of question-answering behavior desired by the questioner, while *respondent-based* models define cooperation in terms of the goals a respondent is trying to achieve in a given interaction. The thesis of this chapter is that in studying CRG we should adopt the respondent-based perspective if our goal is to develop dialogue systems that truly are "cooperative respondents" rather than models of particular kinds of cooperative responses.

## 3.1 The Questioner-Based Perspective

I will say that a model of CRG is *questioner-based* if it characterizes cooperative interaction in terms of how the system's responses affect the questioner in a particular query situation. All judgments of cooperativeness are made from the questioner's point of view; i.e., a response is "cooperative" in a given situation if and only if it is perceived as such by the questioner. Consequently, it is typically not considered important for the reasoning activities of a dialogue system to be defined in terms of abstract concepts such as the beliefs, desires, and intentions of a language user; rather, all that matters is that the program manifests behavior that questioners find acceptable.

A fundamental premise of questioner-based investigations of cooperative behavior is that a questioner in a given dialogue situation has various needs, desires, preferences, and expectations, some of which may be directly evident in his query (and its accompanying utterances), and others of which may require more complex reasoning chains to detect. A questioner may judge a dialogue system uncooperative if it fails to meet his needs, satisfy his desires, accommodate his preferences, and live up to his expectations in a particular query situation. These ideas have given rise to a research program with three main components:

- informally identifying the needs, desires, preferences, and expectations of questioners that are potentially relevant to the processing of a query in a given situation;

- identifying response strategies that can be used to meet the needs, satisfy the desires, accommodate the preferences, and live up to the expectations of a questioner in a given query situation;

- developing models of query situations and designing reasoning procedures on those models that can enable dialogue systems in those situations to respond using the desired response strategy.

I call the three-step methodology suggested by the above research program the *questioner-based approach*; models of CRG resulting from this approach will be called *questioner-based models.* I will discuss next a categorization of questioner-based models and then consider the computational and theoretical consequences of the questioner-based approach. Finally, I will argue that questioner-based models are of little help in building more complex cooperative dialogue systems.

### 3.1.1 Categories of questioner-based models

As noted, the first step of the questioner-based approach to CRG theory involves characterizing (informally) different kinds of response behavior that questioners might want or find desirable from a dialogue system in a given conversation situation. Based on my review of the literature, I have identified three general behavioral goals that past CRG models have had with respect to the questioner:

1. avoiding adverse effects on the questioner;

2. improving a natural-language interface's ease of use;

3. adding useful functionality to a question-answering system.

These goals serve as an abstract categorization of the *motivations* that researchers have offered (or that have been implicit in their work) for dialogue systems to produce the various forms of cooperative question-answering behavior that were studied. I will discuss each goal in turn.

### Avoiding adverse effects

One of the key hypotheses of questioner-based investigations of CRG is that questioners tend to make various kinds of assumptions about the ways in which different knowledge states of a respondent may influence her choice of reply. Questioners consequently interpret the responses given them in light of these assumptions. To borrow a classic example from [Grice 75], suppose Q, carrying an empty gas can, approaches R and asks,

(30) Q: My car has run out of gas. Where's the nearest gas station?

If R were to reply,

(31) R: There's one a quarter-mile up the road, on your left.

Q would probably not feel any urge to ask, as a follow-up question, "Is it open?" Rather, he would assume that R had understood the thrust of the question and would not have answered as she did if she believed the gas station was closed. Furthermore, Q would assume that if R were *in any doubt* as to whether the station was open, she would have indicated as much in her reply (and, perhaps, would have gone on to identify an alternative). As a result, R's simple reply combined with Q's assumptions leads Q to draw a number of conclusions about R's beliefs.

The idea that the questioner's assumptions affect his interpretation of and conclusions drawn from a response is discussed at considerable length in [Joshi 84a]. In that article, the authors examine a scenario in which a questioner asks to be informed how to perform an action $\alpha$ so that he might achieve some goal $\gamma$. They argue that, should it not be possible to execute $\alpha$ (due to, say, the failure of one of $\alpha$'s enabling conditions), it may be insufficient for the respondent to do no more than point that out. They claim that additional response acts may be required *because of the questioner's expectations of the respondent.* For example, the authors suggest that a questioner might expect the respondent to identify an alternative to $\alpha$, if she knew of one that the questioner could use to achieve $\gamma$. Because of this expectation, as the authors contend, a respondent who says only "you cannot do $\alpha$" communicates more than she has explicitly stated. In particular, her statement is said to convey *by omission* that the respondent does not know of (or does not believe there exists) another way for the questioner to achieve $\gamma$.

The fact that what is *omitted* from a response can be as important as (if not more important than) what is *included* in it is particularly troublesome for dialogue systems. It suggests that the absence of certain kinds of language-understanding abilities in a given system may in fact have adverse effects on a questioner as opposed to merely reducing the system's utility. This is partly due to the fact that when people interact with machines using natural language, they tend to make the same sorts of assumptions about the machines' understanding and response abilities as they make when they interact with other people. It is therefore no surprise that a major line of theoretical inquiry has come to focus on finding different kinds of assumptions questioners make that must be accommodated during response generation.

Kaplan observed that some of a questioner's assumptions are accessible to a respondent purely from domain-independent knowledge of the conventions of language [Kaplan 81] (see also Section 2.2.1). He characterized *language-driven inferences* this way (p. 132):

> [Language-driven inferences] are based on the fact that a story, dialogue, utterance, and so on, is a description, and that *the description itself may exhibit useful properties not associated with the thing being described.* These additional properties are used by speakers to encode essential information—a knowledge of language-related conventions is required to understand [natural language].

Kaplan's COOP system employed language-driven inferences to detect (and then verify) the presumptions implicit in a query. The motivation for doing so was to prevent the system from inadvertently confirming a proposition that it knew (or could prove) was false. The three-part hypothesis was that (1) questioners are aware of the assumptions they encode in their queries, (2) they believe that these encoded assumptions are accessible to their interlocutors, and (3) they expect that respondents will notify them of any assumptions that are believed (by the respondent) to be false. A dialogue system that is unable to live up to this third expectation is thus liable to confirm false propositions by omission.

In general, all the work on detecting and correcting misconceptions has been motivated at some level by this desire to prevent dialogue systems from exhibiting behavior that is, from the questioner's point of view, undesirable. Of course, other work besides misconception correction *per se* has had similar motivations. Motro, for example, was partly concerned with preventing users of database-query systems from falsely concluding that correct queries, because they returned "zero" or "nil" as their result, were actually ill formed (Section 2.2.3).[1] And Pollack's plan-inference formalism was an attempt to increase the range of a questioner's beliefs accessible to a respondent, ultimately to permit respondents to detect misconceptions and modify their responses accordingly (Section 2.4.1, page 36).

### Improving ease of use

Several models of CRG have had the goal of making dialogue systems easier in some sense to interact with. For example, questioners are seen as preferring systems that do not unnecessarily prolong information-retrieval dialogues. Some researchers have therefore tried to exploit both domain-independent characteristics of questions and domain-specific knowledge to enable systems not only to answer questions, but also to provide additional related information in their responses.

In the literature review we saw three efforts (Gal's in 2.2.4, Motro's in 2.2.3, and Wahlster *et al.*'s in 2.2.6) in which one of the goals of the system was to provide responses that were said to "anticipate likely follow-up questions". One oft-repeated point has been that questioners tend to ask questions of the form, "is it the case that $P$", not because they are interested in the truth value of $P$ *per se*, but because they are interested in the answer to a related wh-question whose felicity depends on the truth value of $P$. The general research problem involves finding a principled connection between the yes/no query, its answer, and the follow-up question (or questions) that the questioner could be expected to ask given that answer. Gal showed that anticipatory question-answering behavior may be applicable to other kinds of questions as well (such as "how many" questions), which suggests that some general question-independent underlying reasoning process must be involved.

[1] This might be an example of a meta-misconception: a misconception that one has a misconception.

Models of anticipatory question answering have varied widely in their details. In Gal's system, domain-specific integrity constraints were indexed during query processing; the theory of what information a questioner might find useful in addition to the answer was encoded in the indexing strategies. In Motro's system, failing queries were generalized until a non-failing query was found; this non-failing query was then treated as a likely follow-up question and its answer was provided. HAM-ANS augmented its responses with the values of case-slots that were filled as a side effect of determining the answer to the query. Others have shown that more general inference mechanisms may be needed to determine appropriate additions. Allen, for example, observed that it seems necessary to reason about the questioner's goals in order to determine the most appropriate addition [Allen 83]:

(32) Q: Is John a senior?
(33)a. R: No, he's a junior.
b. R: No, but Sam is.

The choice between (33)a and (33)b hinges upon whether R believes that Q is interested in John's class rank, or whether he needs to know the name of someone of senior standing.

All these efforts are motivated by the recognition that dialogue systems that are able to provide additional information on their own initiative will be found to be considerably more pleasant to use than those that require each "obvious" follow-up query to be posed explicitly. This motivation is quite different from that of the work on misconception correction (and related behavior). Here it is not that a user could or would be harmed by a system's failure to volunteer information, but rather that systems able to "over-answer" questions will appear to communicate more helpfully and in a manner more in accord with the expectations people have of the question-answering behavior of other people.

### Adding useful functionality

The third group of questioner-based models has focused on enabling systems to exhibit useful response behavior of which a questioner might not ordinarily expect them to be capable. Mays's development of a logic to support the offering of monitors by database-query systems is perhaps the best example (Section 2.2.5). The principle is that a user might not be aware that a database system could keep track of changes to the database and perform actions when certain conditions became true. That, coupled with the notion that a questioner who inquires about the truth status of a (currently false) time-varying proposition $P$ might be interested in being notified when $P$ becomes true, inspired Mays's research into a mechanism for distinguishing potentially useful monitor offers from useless or nonsensical ones.[2]

Luria's system for identifying possible plan failures (Section 2.3.4) is another attempt at adding useful functionality to dialogue systems. The idea is that when responding to "how do I do X" questions, it is often in the questioner's interest to inform him of conditions he might have to rectify before the plan that the system gives him can be executed. In other work, research on constructing examples and integrating them into responses to requests for on-line help has been reported by Rissland [Rissland 84]. Chin's UCEGO is also able

[2]Mays had little to say, however, about how to distinguish *relevant* from *irrelevant* monitor offers.

to add examples to the responses it gives [Chin 88a] (and see also Section 2.3.3). Finally, I have already pointed out that McKeown's work on TEXT was originally conceived of as part of a larger effort to make it unnecessary for users of database-query systems to have to know the details of the database system's structure and terminology (Section 2.2.7).

#### Summary

I have distinguished three general behavioral goals that past models of CRG have attempted to achieve. Each goal defines a *class* of specific cooperative goals that systems have been designed to achieve. "Avoiding adverse effects", for example, circumscribes a class of more specific goals pertaining to the prevention of various kinds of adverse effects on the questioner. Correcting a presumption failure (or any misconception) can be conceived of as but one of many ways of avoiding an adverse effect (that of being misled) on the questioner.

There are two important problems to be recognized in this regard. First, in no case have the *specific cooperative goals* of a CRG system been precisely stated. In other words, the actions performed by the various systems were all done for reasons that were external to the systems. Second, as a consequence of the first, no *domain-independent motivation* for a given response action was ever provided. That is, not only were the *goals* of the systems extra-theoretical, but the *reasons for adopting those goals* were extra-theoretical as well. These ideas will be explored further in the next section.

### 3.1.2 Consequences of questioner-based approach

#### Computational consequences

I claim that the questioner-based approach to investigations of CRG has led to the development of computational models of what might be called *cooperative response situations*—situations in which a particular response or kind of response is said to be cooperative. The systems that have been implemented to validate the results of these investigations can be viewed abstractly as functions from carefully restricted question-answering contexts to responses claimed (always on intuitive grounds) to be cooperative in those contexts.

The general technical approach goes roughly like this: after analyzing a set of examples—either constructed or taken from transcripts of naturally occurring dialogue—in a particular domain, the researcher identifies a kind of response behavior desired from some hypothetical dialogue system. Next, the question-answering situation is specified. This involves selecting the range of questions or question types to be handled by the system and defining all the knowledge sources (and their contents) available to the system at query-processing time, such as concept taxonomies, database schemas, factual knowledge, plan libraries, user models, etc. This collection of knowledge sources will be referred to as the *domain knowledge base.* In some cases, domain knowledge bases have possessed properties that were essential to the needs of the particular response-processing task, e.g., the domain model augmented with an "object perspective" mechanism in ROMPER (Section 2.3.1) and the temporal database axiomatization in Mays's theory of monitor offering (Section 2.2.5).

The third step involves specifying the input to the response-computing component of the dialogue system. This input precisely defines the conditions in effect at the start of response processing. I will call this input the *query model.* In COOP, for example, the query model consists of an MQL representation of the input query [Kaplan 82]; in ROMPER, it is a description of a particular object-related misconception. In Allen's system, the query model comprises a logical-form representation of the information requested by the questioner (e.g., a WANT to KNOWIF there is a train to Windsor) [Allen 83]. Thus the query model represents the knowledge state of the dialogue system with respect to a given query just prior to response selection. There is usually some (often unspecified and/or unimplemented) amount of processing interposed between the actual natural-language input and the response-computation machinery. The general architecture of the resulting dialogue system is shown in Figure 3.1.

Once the domain knowledge base and query model have been specified, the rest of the research concerns the design of the component that is actually responsible for computing the system's response (labeled "CRG system" in Figure 3.1). At an abstract level, the CRG system consists of two components: the *cooperative-response trigger* and the *response-generation procedure.* (Note that research systems have often focused on the design of only one of these two components; ROMPER is an example of a system that focused on the generation procedure.)

The *cooperative-response trigger* (labeled "CR trigger" in Figure 3.1) is a decision procedure over the query model that determines when the particular form of cooperative behavior under study should be exhibited in the response. To illustrate, we will use Kaplan's COOP as a case study. The desired response behavior is the act of producing a corrective indirect response. The domain can be any common database-query domain; this discussion will assume a database of department-store inventory. Questions of interest include those that (explicitly or implicitly) predicate over the results of various kinds of set intersections and restrictions. For example, the query in (34) requests that the system list the members of the set resulting from the intersection of two restricted sets: the set of departments restricted to those that sell coffee makers and the set of departments restricted to those that sell coffee filters.

(34) Q: Which departments that sell coffee makers also sell coffee filters?

The query model contains a representation of the user's query and the answer to it as returned from the database.

In COOP, the cooperative-response trigger performs two tests. First, the answer returned from the database is checked. If it is "zero" or "nil," the query in the query model is decomposed into its set components, and each of those sets is tested against the database for emptiness. If any of these sets are found to be empty, e.g., in Example (34) there are no departments at all (rather odd, but not inconceivable) or no departments that sell coffee makers, then the decision procedure returns *true*, indicating that the current query situation demands a corrective response.

The *response-generation procedure* (labeled "RG proc" in Figure 3.1) is activated by the cooperative-response trigger and proceeds to construct and generate the cooperative response. This can be either a simple or a complex task, often depending upon what was involved in the triggering test. In COOP, the response-generation procedure uses data generated as a by-product of the cooperative-response trigger's processing: the set of sets embedded in the query that were found to be empty.[3] The actual natural-language

[3] To be specific, the set of (so-called) *least-failing* descriptions embedded in the original query.

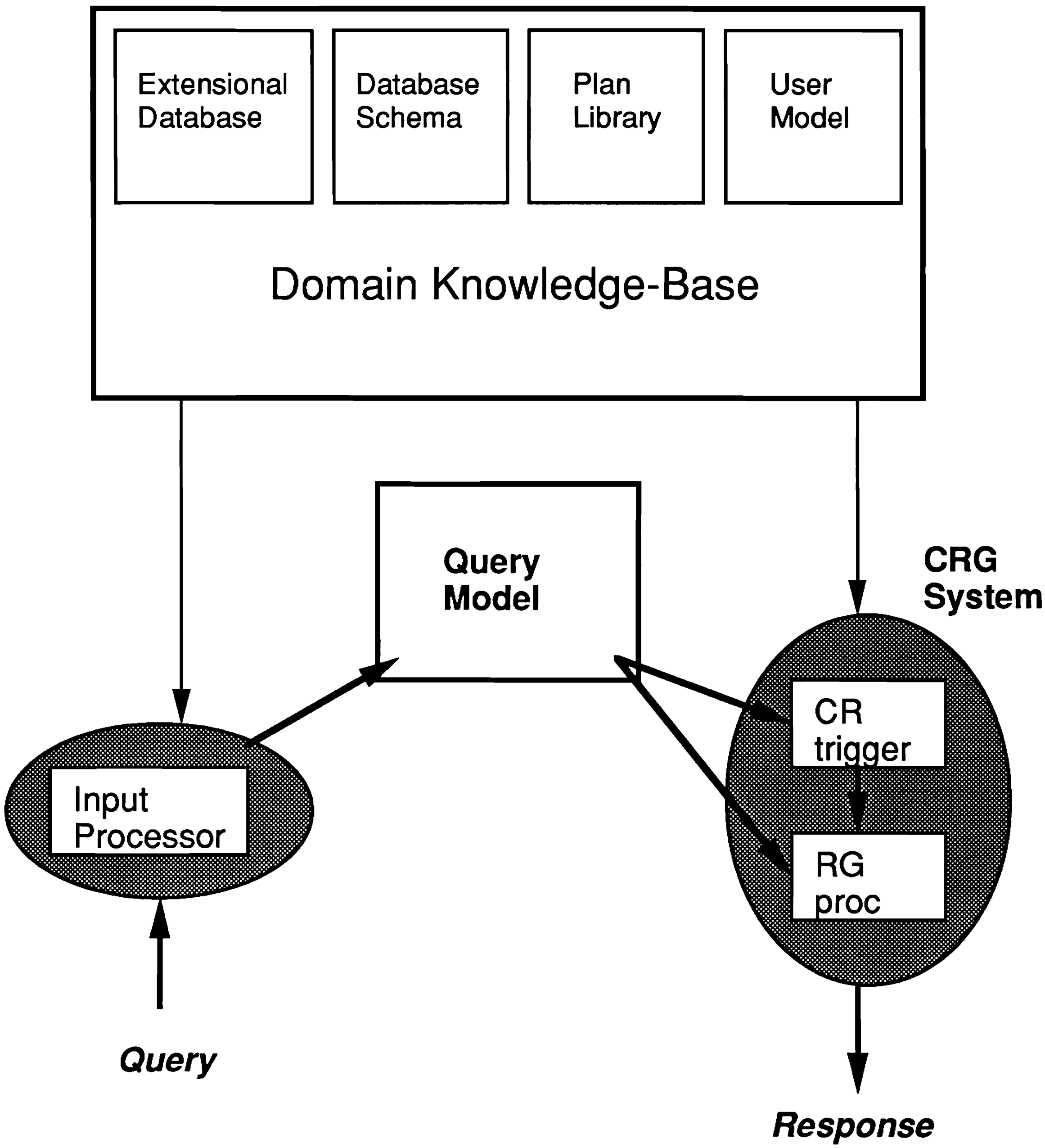


Figure 3.1: Cooperative Response Situation Architecture

response produced is then a modest transformation away, involving little more than filling in and generating the pattern, "I don't know of any $x$", substituting each element of the least-failing set of descriptions for $x$, e.g., "I don't know of any *departments that sell coffee makers.*"

Note that researchers have tended to factor out the details of generating natural language, or have used template-filling or other simple methods for translating from the system's internal representations to natural language. The exceptions include ROMPER and TEXT, both of which focused almost entirely on the language-generation task (TEXT more so than ROMPER), and the UCEXPRESS component of the UNIX Consultant [Wilensky 88].

The diagram in Figure 3.1 should be viewed as an abstraction of the way questioner--based models of CRG have processed queries and produced responses. The combination of query model and domain knowledge base constitutes a model of a particular question-answering situation. The CRG system (the combination of cooperative-response trigger and response-generation procedure) acts as a machine that, when placed in that situation, produces the prescribed "cooperative" response.

With respect to this overall approach, there are two significant points to be made. First, the responses, or at least the kinds of behavior desired in responses, have been "fixed targets" in each case. Once it was decided what the questioner needed or wanted in a particular situation, the computational problem boiled down to constructing the machinery needed to "hit the target". Thus the computational mechanisms involved have been devoid of any notion of "cooperative behavior". Second, characterizations of situation models (query model plus domain knowledge base) have varied widely from system to system. Disjoint sets of questions have been considered, knowledge sources have been defined differently and/or differed in the special properties they possessed, and both the amount and nature of the processing (implicitly) required to construct the query models presented to the CRG systems have varied. This is because the focus was solely on building systems that manifested some desired behavior, thus leaving the researchers with many degrees of freedom in setting the problem up. I suggest that it is largely for this reason that the collective theoretical results to date are so incompatible.

### Theoretical consequences

I claim that the theories of cooperative behavior yielded by the questioner-based approach are uniformly *descriptive.* That is, they describe in informal terms what it is that makes a particular response "cooperative" in a given situation and then concern themselves with the task of generating it. They rarely if ever attempt to specify the factors that make a given response cooperative, and dialogue systems to date certainly have never been grounded on any theory of what it means to cooperate.

Consider the act of correcting a misconception. Why and under what conditions should this be considered a cooperative act? Several systems have been designed to correct misconceptions, but they never had access to—were never able to reason about—*why* they were correcting a misconception. For example, no theory of misconception correction has ever involved explicit reasoning about, say, the possible effects on the questioner of continuing to hold a particular invalid belief. Such a theory would at least tell us something about the conditions under which it is *not* necessary to correct a misconception (something no existing theory does). Instead, the context was always contrived so that the need for the corrective response was always "obvious" in the given context and never had to be considered explicitly. But it is exactly those elusive features of the interaction causing the corrective response to seem "obviously appropriate" that are essential to a theory of cooperation.

Because the desired response behavior was decided in advance (and the question-answering situation set up with that in mind), researchers were able to make a kind of *single-fault assumption* with respect to the response: that is, it could be assumed that the question-answering context would never motivate any kind of response behavior other than the one under study. Consider again the body of work on correcting misconceptions. The response situation models were always chosen so that no other act besides the corrective response could possibly be appropriate. But when we examine more realistic models of discourse, we see this to be an artificial restriction. Consider the example shown in (35), taken from the EMACS transcripts.

(35) Q: Do I want a mark at the beginning and at the end of the block?
R: For right now, just go ahead and set a mark any old place. (Just to see how it works.)

R is engaged here in teaching Q how to use "region" commands in the EMACS text editor. In order to change a block of text to lowercase, an EMACS user must set the "mark" (of which there is only one) at the beginning of the block, then move the "point" (corresponding to the cursor) to the end of the block. At this time, a region command (like "lowercase region") can be executed. In the discourse leading up to the example (as recorded in the transcripts), R has been trying to show Q how to set the mark. The query shown in the example comes after R has described the "set mark" command and has instructed Q to execute it. Q's query indicates a misconception: he seems to think that there is more than one mark. But R does not correct that misconception.

I am not claiming that R in this example did, in fact, notice the misconception and decide not to correct it (that would be a highly speculative claim indeed).[4] However, I *am* claiming that the example illustrates a situation in which more pressing concerns might very well lead a respondent to overlook a misconception. Since R is engaged in a teaching task, she might decide that Q will figure things out for himself later as the dialogue proceeds, or perhaps she might conclude that correcting the misconception now, given the current state of the discourse and her beliefs about Q's understanding of the task, would be difficult and/or distracting. The point is that the existence of a misconception in and of itself is not sufficient reason in general for carrying out a correction. An approach that only describes a situation and stipulates that a particular form of behavior is cooperative will not tell us anything about situations in which that behavior is unnecessary or inappropriate. In short, such an approach will not really tell us why a particular response act is cooperative.

Ultimately, the questioner-based approach has given rise to theories of cooperation that are distinctly "Pavlovian" in the sense that the input situation—represented by the query model—serves as the "stimulus" driving a predetermined kind of response. It is for this reason that the nature of cooperative interaction is hardly better understood now than when research in this area began. Instead we have accumulated informal characterizations

[4]In a subsequent exchange in the recorded transcripts, Q makes a reference to "the first mark." It is at this point that R explicitly corrects the misconception, but only in the context of a larger response. The relevant fragment is shown on page 58, in example (36).

of user needs, desires, preferences, and expectations to be addressed in different, literally incomparable situations.

### 3.1.3 Questioner-based models: ultimately the wrong approach

I have characterized questioner-based models and the questioner-based approach to CRG theory, and have discussed the computational and theoretical consequences of that approach. I now want to argue that we must abandon the questioner-based approach if we intend eventually to build dialogue systems that are not just models of particular kinds of cooperative responses but truly are "cooperative respondents".

#### The toolbox metaphor

My main point here is that for system designers developing new natural-language applications that must be able to provide cooperative responses, questioner-based models provide no guidance except in those cases where an exact match can be found between an existing questioner-based CRG model and a query situation that the application must handle. Unfortunately, due to the artificial restrictions that have been placed on the question-answering situations handled by questioner-based models, such precise matches are likely to be hard to find. Rather, natural-language systems in realistic domains can be expected to face query situations that will cross the artificial boundaries that have been established by previous CRG models. Consequently, such systems will need to be able to *integrate* their cooperative abilities when formulating their responses. But the incompatibility of questioner-based models makes solving this "integration problem" difficult, if not impossible.

Why is that the case? As we saw in the literature review, past research has succeeded in identifying and labeling a great variety of forms of cooperative response behavior. Many of these pertain to misconception correction: examples include correcting so-called "object-related" misconceptions, "plan-related" misconceptions, or (intensional/extensional) presumption failures. Other examples include "over-answering" questions, offering monitors, detecting and removing "knowledge obstacles" in plans, and so forth. One might be tempted to view these forms of behavior as analogous to tools in a kind of "cooperation toolbox".

But this *toolbox metaphor* is of limited usefulness. To its credit, it helps us understand the questioner-based approach in a general way: each researcher defined a new "tool for cooperation" and applied it to query situations that were set up to require that tool and no other. But it misleadingly suggests that we can solve the integration problem and produce more realistic kinds of responses by applying several tools at once in a given query situation. The problem with that approach is that the tools simply are not designed to be used in concert. For example, McCoy's ROMPER system for correcting object-related misconceptions assumed that it would be called upon to generate the system's entire response. But as the following example from the EMACS transcripts shows, a misconception-correction may be only a small part (shown in boldface) of a response:

(36) Q: Wait—I set the first mark at the bottom of what I wanted so I didn't want to delete anything below that. I tried typing ESC-G a few lines below but saw no response.

R: Hmmm. Let's do that again. Set the mark **(there is only one mark—THE mark)** move over one word or line, and type ESC-G. The point (cursor) should move back and forth. Yes?

The example shows that in more natural discourse, cooperative acts such as misconception corrections must be integrated in some way into the respondent's overall response goals. The example also suggests that any attempt to generalize the response into a kind of cooperative response situation—by applying the questioner-based approach—would be completely *ad hoc.*

Of course, the assumption made by ROMPER—that it was wholly responsible for computing the response—is shared by every other model of CRG to date. I call this the *whole-response assumption.* It is the assumption at the very heart of the questioner-based approach, and it has led to theories that cannot be integrated.

The questioner-based approach must be abandoned if we are to make progress towards understanding cooperative response behavior. I have presented two arguments for this conclusion: first, more capable cooperative dialogue systems will need to integrate their cooperative abilities in their responses, and the questioner-based approach is inimical to such integration. Second, by focusing on the design of systems that only *manifest* cooperativeness, questioner-based models provide precious little insight into what it means for a respondent to "be cooperative".

### Caveat

I have intentionally taken a strong position here in painting a diversity of research with the label "questioner-based model". I want to emphasize that I do not believe that many past research efforts were truly concerned in any serious way with probing the nature of cooperative behavior. In fact, I doubt that *any* of them were. Rather, they were interested in mechanizing some of the many forms of useful and helpful response behavior found in real-world domains and question-answering situations. From that perspective, they have all been useful. The work in plan recognition (and in intention recognition more generally) has been particularly helpful in elucidating the representations and reasoning mechanisms that are essential to cooperative communication. My point, however, is that the artificial restrictions that have been imposed on models of question-answering situations have had the unfortunate effect of obscuring rather than illuminating the basic principles of cooperation. This has in turn made it very difficult to see how past efforts might be extended, generalized, and used as building blocks in the construction of more capable cooperative natural-language systems.

## 3.2 The Respondent-Based Perspective

A model of cooperative response generation is *respondent based* if it characterizes both the knowledge and the decision-making process *of the respondent* that together account for her choice of cooperative response in a given query situation. Respondent-based models attempt to define *the respondent's role as a cooperative conversational partner.* When viewing CRG from the respondent's perspective, the theorist analyzes examples of cooperative response behavior in an effort to uncover the general principles of reasoning that motivate the respondent's selection of the form and content of her response.

In contrast, questioner-based models yield techniques by which dialogue systems can produce responses that lend those systems the appearance of behaving cooperatively. Those techniques, however, need not be (and have not been) grounded on principles of cooperative behavior and reasoning. The theoretical interest here is in *how* responses of a desired form can be produced rather than in *why* respondents might choose those forms of response.

This section elaborates the characterization of the respondent-based perspective and examines several issues and examples that are best understood within a respondent-based theoretical framework.

### 3.2.1 Responses reflect goals

At the heart of the respondent-based perspective is the view of language as action that forms the foundation of *speech-act theory* [Searle 71]. In short, *responses reflect the respondent's goals.* That is, the utterances composing a natural-language response represent the observable result of the respondent's execution of a plan she had formed to achieve particular goals. (Those goals, however, may not always be immediately apparent in the response.)

Consider the following example, taken from the UNIX transcripts.

(37) Q: Is there a way to send mail to ucbeuler from ucbcory?
R: Yes, it is letter y on the Berknet. So mail user@y.CC. If you have further problems with it, mail to serge@cory. He is the euler system manager.

The response in (37) might be modeled at the surface level as an attempt to achieve these five goals:

1. Q *believe that* there is a way to send mail to ucbeuler from ucbcory ("Yes...").

2. Q *believe that* the letter on the Berknet that corresponds to ucbeuler is 'y' ("...it is letter y on the Berknet...").

3. Q *believe that* because the letter on the Berknet that corresponds to ucbeuler is 'y', mail to users on ucbeuler can be addressed using the syntax user@y.CC ("...it is letter y on the Berknet, so mail user@y.CC").

4. Q *believe that* if he has difficulty sending mail to users at ucbeuler, he should send mail to serge@cory asking for assistance ("If you have further problems with it, mail to serge@cory").

5. Q *believe that* asking assistance from serge@cory is reasonable because serge is the system manager for ucbeuler ("He is the euler system manager").

These are the goals that we might think of as lying "just beneath the surface" of the response, that is, the goals that the particular linguistic expressions are being uttered in an attempt to achieve. At a deeper level, the response might be viewed as an attempt to achieve two more abstract goals:

1. Q *know that* it is possible to send mail to ucbeuler from ucbcory.

2. Q *know how* to send mail to ucbeuler from ucbcory.

While the attempt to achieve the first goal is straightforwardly evident in the surface form of the response (R's answer of "yes" to the literal question), the attempt to achieve the second takes several utterances. In order to achieve that second goal, R must—as a subgoal—cause Q to know the appropriate command syntax. In the process of describing the command, R finds that she must *explain* the appearance of the 'y' in the command and also must *suggest* an action for Q to take in case of difficulty. R finally decides that her suggestion that Q send mail to serge@cory must be *justified.*

I cannot claim that this is the actual goal structure underlying R's response, for that would be pure speculation. Rather, I am suggesting that R had certain goals in mind to achieve in her response, and showing how her actual utterances might arise from her attempt to achieve those goals.

It is the central hypothesis of this dissertation that:

> In order to build dialogue systems that are to be more generally capable of acting as cooperative partners in a conversation, we must view their responses as actions planned to achieve one or more goals.

I want it to be clear that the above is a *hypothesis*, not a claim. It is not my intention in this dissertation to try to prove the hypothesis. However, by means of the discussion and examples presented herein I intend to argue that the hypothesis is both reasonable and useful from the point of view of CRG system design.

The hypothesis clearly embodies the respondent-based perspective. As I have said, respondent-based models of CRG attempt to characterize the respondent's reasoning in cooperative interaction. This means that the respondent must play an explicit role in any theory of cooperative communication; according to the hypothesis, she does, being the one who selects her response goals and forms plans to achieve them.

### Comparing the perspectives

Now that we have formulated our hypothesis, it is an appropriate time to compare the two perspectives on CRG and determine precisely what distinguishes them. My claim is that in all questioner-based models, the goal (or goals) that the system is attempting to achieve is left unspecified. That is, although the behavior that the systems exhibit are all attempts to achieve goals (in line with the hypothesis), those goals have never been defined or formalized (or, at least, never in any general way). At best, the goals of the systems have been "defined" using natural-language descriptions.

This point is best understood by example, for which we will again consider Kaplan's COOP system. COOP was described as being able to produce corrective indirect responses in order to correct false presumptions detected in queries. When COOP determined that a given query presumed the non-emptiness of some set $S$ of database entities, rather than providing the zero or nil query result to the user, it would reply, "I don't know of any $T$s", where $T$ is the type of the elements of $S$.

Using an intuitive logic of belief,[5] we might represent the system's response as an attempt to achieve goals of the form:

[5] See [Hintikka 62] for one formalization of a logic of belief.

```
(BEL Q (BEL R (EMPTY (THE S:SET (SETOF S T)))))
```

That is, by uttering "I don't know of any $T$s", R (that is, COOP) intends that Q (the user) believe that R believes the set $S$ of all $T$s is empty. I will call this the *surface goal*, the goal that the surface linguistic expression is intended to achieve. The surface linguistic expression actually used may be one of many different ways of attempting to achieve the surface goal.

Now we might ask, what is the formal connection between the surface goal and COOP's cooperative purpose of correcting presumption failures? First we must understand at an abstract level what operation COOP's machinery is actually performing. I suggest that this operation is actually a process of bringing to COOP's attention two beliefs of the form:

```
(BEL R (EMPTY S))
(BEL R (BEL Q (NOT (EMPTY S))))
```

for $S$ denoting a set of database entities of type $T$ (e.g., "French students", or "coffee makers"). Once COOP acquires such a belief, it essentially adopts the following goal:

```
(GOAL R (BEL Q (EMPTY S)))
```

I will call this the *response goal*, to distinguish it from the surface goal. The response goal is the most general goal that underlies the response. A relationship analogous to *generation* [Goldman 70, Pollack 86] exists between response goals and surface goals: each response goal is achieved by means of the satisfaction of some set of surface goals. In COOP, each response goal gives rise to exactly one surface goal, and these surface goals are then achieved using a fixed utterance form.[6]

To summarize, COOP's abstract reasoning behavior can be modeled as the following four-step procedure:

1. Analyze query: identify all sets $S$ such that the query presumes `(NOT (EMPTY S))` but COOP believes `(EMPTY S)`.

2. For each such $S$ (of elements of type $T$) such that it is not a subset of any other empty set, adopt response goal: `(GOAL R (BEL Q (EMPTY S)))`.

3. To achieve each response goal, satisfy surface goal:
   `(GOAL R (BEL Q (BEL R (EMPTY (THE S:SET (SETOF S T))))))`

4. Use utterance form "I don't know of any $T$s" to satisfy surface goal.

Somewhat more generally, the reasoning process can be described this way:

1. Identify an issue to be addressed in the response (Q incorrectly believes that set $S$ is non-empty).

2. Adopt the goal of addressing that issue by generating a correction (cause Q to believe that the set $S$ is empty).

3. Form and execute a plan for that goal: cause Q to believe that COOP believes the set $S$ of $T$s is empty; realize using a simple assertional linguistic expression.

[6]There is an implicit assumption that when COOP asserts $P$ (thus indicating its belief that $P$), Q will adopt $P$ as one of his own beliefs.

We are now in a position to make some interesting observations about both COOP and the distinction between questioner-based and respondent-based theories.

Most importantly, COOP had no declarative representation of the issue it was addressing, the goal it adopted as a way to address that issue, or the plan it formulated for that goal. Instead, there was a direct procedural connection between the analysis of the query and the corrective response that the system generated. It is for this reason that I consider COOP to be a questioner-based model. The purpose of the system is to provide a desired corrective response when one particular analysis of the query detects the presence of certain conditions. All the reasoning that a *respondent* might perform is factored out of the model. This of course raises questions concerning the sorts of reasoning that a respondent might want or need to perform.

COOP had no understanding of what a "false presumption" is or why such things might need to be corrected. We see this in the model I developed above: once COOP notices that

```
(BEL R (EMPTY S))
```

AND

```
(BEL R (BEL Q (NOT (EMPTY S))))
```

it reflexively adopts the goal of causing Q to believe that the set $S$ is empty. But this behavior is reasonable only because of the restricted nature of the interaction. For one thing, it is never possible in the domain of database querying for R to have incorrect knowledge.[7] Consequently, given the above-mentioned belief, R need never consider whether she is mistaken in believing that the set $S$ is empty. Moreover, because the whole purpose of dialogue in this domain is to inquire about sets and their members, it can be assumed that whenever it is discovered that Q incorrectly believes a particular set to be non-empty, he would want to be informed of his error.

Besides factoring out all the reasoning leading to the decision to correct the questioner's incorrect belief, COOP uses a canned plan to achieve that goal of correction. COOP has no way to decide whether that plan is a good one with a reasonable chance of success, nor can it decide to, e.g., justify or explain why it believes Q is mistaken. All of these decisions have been made in advance and hard-wired into the system.

The criticisms that I have applied to COOP can be equally applied to the other research efforts in CRG. All fail to declaratively state the systems' goals in responding in the ways they do, and consequently all fail to define a principled connection between their goals and the responses that are generated. For two more examples, consider that ROMPER's misconception-correction goals are not defined, nor is it shown how the correction strategies, when executed, realize those goals; in Allen's plan-recognition system, there is no representation of what a knowledge obstacle is, why it should be eliminated, or how goals of obstacle elimination might be achieved.

The point of the preceding discussion is not to devalue the contributions of prior research. The work being described here would not have been possible if it were not for the existence of such a large and diverse body of scientific results in the area of CRG. Rather, my point is this: cooperative responses generated in natural conversational settings (e.g., interactions between users and consultants) often show that respondents attempt to achieve more than one goal at a time in their replies. If we want to be able to build dialogue systems that have this ability, theorists must begin to identify and specify declaratively the

[7] Pollack has called this the *correct-knowledge assumption* [Pollack 86].

kinds of goals cooperative respondents attempt to achieve in their responses, the conditions of the conversation that lead them to adopt those goals, and the principles of reasoning that connect the adoption of their goals to the responses they generate. For too long the focus has been on techniques for achieving ill-defined or unspecified goals. We must probe more deeply than that.

#### Toward modeling the respondent's goals

Although this is the first effort to focus directly on the principles and processes by which cooperative respondents reason, adopt and pursue goals, it is not the first to recognize the necessity of modeling the respondent's goals. In the UCEGO system, Chin models the respondent's goals directly [Chin 88a]. But as I argued in Section 2.3.3, Chin's foci were on identifying general categories of goals that participate in cooperative reasoning (themes, foreground and background goals, sub-goals, and meta-goals) and on showing how to model them uniformly using a single computational mechanism (the if-detected daemon). His selection of implemented goals, while intuitively plausible, was not based on any developed theory of interaction in the domain, and as a process model he used an unconstrained blackboard architecture. In contrast, the present dissertation aims to develop an initial outline of the process of cooperative reasoning as a step toward the formulation of a true theory of cooperative dialogue.

Cohen and Perrault also advocate a respondent-based view in their plan-based theory of speech acts (see the review in Section 2.4.1). As I have discussed, however, the speaker goals modeled in that work are at a level of detail that places them beyond most of the concerns of cooperative dialogue. In developing the respondent-based perspective, I am arguing that theories need to be developed both of the kinds of high-level goals that are called themes in UCEGO and of the process by which those goals are adopted and pursued.

### 3.2.2 A respondent-based research program

At the beginning of this chapter I raised the question, why do we in computational linguistics study cooperative response generation? I have long felt that this is an important question to which inadequate attention has been paid. When one surveys the relevant literature, only one answer (and it is an implicit one at that) comes to mind, namely, to circumscribe and model distinct forms of cooperative response behavior. While some interesting and useful results have come from this line of inquiry, they have, as noted herein, generally suffered from a lack of comparability and compatibility.

I suggest instead that research in CRG should have two separate but related goals: first, to develop natural-language dialogue systems that are more generally capable of cooperative communication, and second, to develop a principled understanding of what it means to be cooperative. The arguments in Section 3.1 were intended to show that the questioner-based research program does not lead to progress toward those goals. My claim is that in order to achieve them we must adopt the respondent-based theoretical perspective.

Like the questioner-based perspective, the respondent-based perspective implies a research methodology. That methodology is suggested by the analysis of COOP discussed in the previous section. If we look carefully at that analysis, we can identify five aspects of knowledge that participate in the process of cooperative reasoning:

1. the goals the respondent is attempting to achieve by means of her response;
2. the conditions of the conversation that motivate the adoption of those goals;
3. the knowledge and inference mechanisms needed to detect and reason about those conditions;
4. the principles of cooperative reasoning that connect detected conditions to response goals;
5. the response strategies that are useful for achieving response goals.

A respondent-based research program aims to deepen our understanding of these components of cooperative reasoning. The particular program I propose thus consists of these five elements:

1. developing declarative definitions, in a consistent formal notation, of the kinds of cooperative response goals respondents attempt to achieve;
2. identifying the conditions of conversations that may lead respondents to adopt their various possible response goals;
3. identifying the knowledge and reasoning resources needed to analyze models of conversations and determine what sets of conditions hold;
4. developing models of the reasoning processes connecting conversation models to the adoption by respondents of their response goals;
5. developing models of the strategies that are useful for achieving various response goals and specifying their conditions of applicability.

I will explain each of these elements briefly in turn.

### Developing declarative definitions

I have already discussed at length the concern that questioner-based CRG systems produce behavior intended to achieve response goals that are never defined in the associated theory. If we are to make progress toward more generally-capable cooperative dialogue systems, we need to develop precise specifications of those goals. For example, we need to define the goals that underlie misconception correction, the goals that underlie obstacle elimination, and the goals that underlie over-answering behavior. I conjecture that if we were to examine all the work on CRG to date, we would discover that many systems share at an abstract level the same response goal, varying mostly in the conditions under which they adopt that goal, the machinery needed to detect those conditions, and the strategies applied in pursuit of the goal.

### Identifying conditions of conversations

As I have suggested, there is a distinction between the *response goals* that a given CRG system might want to achieve in a given query situation and the *conditions of the conversation* that motivate the system's adoption of each of those goals. For example, COOP's misconception-correction goals are motivated by presumption-failure conditions—conditions arising from the detection of false presumptions of the user's query. The proposition that "query $Q$ falsely presumes TRUE($P$)" (for $P$ a proposition of the form "the set $S$ of $T$s is non-empty") can be viewed as an abstract condition that may or may not hold in a given query situation; its actual truth value in a particular situation (as perceived by the respondent) may serve to fully or partially motivate the adoption of one or more response goals. In COOP's case, I argued that the truth of such propositions motivates the adoption of goals to cause the questioner to believe FALSE($P$). Therefore, I claim that in order to better understand CRG, research must aim to develop better specifications of the conversational conditions that influence respondents' choice of cooperative response goals.

### Identifying knowledge and reasoning resources

Perhaps one of the more valuable contributions of prior research has been the identification of some of the knowledge and reasoning resources needed to *detect* the conditions that motivate cooperative response goals. For example, Allen's plan-recognition theory [Allen 83] can be viewed as partially defining the knowledge—knowledge about plans and actions—and reasoning resources—the plan-inference mechanism—that permit dialogue systems to detect knowledge obstacles. The presence of a knowledge obstacle then functions as a condition that might motivate a decision to provide information in the response intended to eliminate that obstacle. Research must continue on techniques for detecting conditions that motivate cooperative response behavior; it should, however, be more meticulous about developing general specifications and representations of those conditions.

### Principles linking conditions to response goals

As I have tried to show, there is an important phase of reasoning and decision making that comes between the detection of goal-motivating conditions and the respondent's formulation and adoption of her response goals. The questioner-based approach has made this phase all too easy to overlook or trivialize; this is the phase that must be based on real principles of cooperative behavior. This is the phase during which the respondent must decide, based on her knowledge of the truth status of various goal-motivating conditions, which cooperative response goals can and should be adopted.

As argued earlier, for example, the detection of a misconception—the condition corresponding to the respondent's perception that the questioner holds a misconception—may not always demand that a correction be made part of the response. Other considerations may come into play in the process of deciding whether to issue a correction. The principles of reasoning according to which such decisions are made constitute a body of knowledge distinct from both that of the goal-motivating conditions and that of the goals themselves. It is in this body of knowledge, more than in any other, that we will ultimately find a theory of cooperative behavior. Modeling it will be a difficult task, but it will be essential as the dialogues that systems engage in become increasingly complex.

### Modeling response strategies

For a given cooperative response goal there may be different ways of achieving it in a particular question-answering situation. For example, the different questioner-based misconception-correction systems can be viewed as implementations of different strategies (whose conditions of applicability are functions of the misconception type) for achieving goals of correction. More than one strategy at a time may be useful for achieving a given response goal, and there may be non-trivial knowledge and reasoning processes required to select the best strategy. Developing precise specifications of these goal-achievement strategies and their conditions of applicability will be a useful direction for further research.

### Summary

In this section it was suggested that research in cooperative response generation should be directed toward two separate but related goals: the development of dialogue systems that are more generally capable of cooperative communication, and the formulation of a theory of cooperative behavior. I discussed five aspects of knowledge that I claim participate in the cooperative reasoning process, and outlined a general program of research aimed at deepening our understanding of those aspects of knowledge.

## 3.2.3 Some issues in the development of cooperative dialogue systems

If we follow the general research program that I have outlined and try to develop dialogue systems that are cooperative respondents and not just models of cooperative responses, we will encounter many complexities of cooperative interaction that can be neither studied nor modeled in a questioner-based framework. Several of these will be considered next. I will not be attempting to propose any solutions; I am merely discussing some observations about a few interesting examples of cooperative behavior that would be worth investigating further in a respondent-based framework. The following points will be discussed:

- A respondent may choose not only whether to cooperate but also to what degree she will cooperate.
- A cooperative respondent may be able to distinguish her obligations from her options in a given response situation.
- A respondent's ability to cooperate is a function of the breadth and depth of her domain knowledge.
- A respondent's choice of response may be influenced by her evaluation of the questioner's goals and plans.
- A respondent's choice of response may be influenced by both the mode of communication and the amount of effort required to convey information.
- A respondent's own goals may affect her choice of response.

Taken together, the examples discussed in this section provide strong evidence for the hypothesis that responses must be viewed as actions planned to achieve one or more goals.

### A matter of degree

In the past, cooperativeness has been treated as a discrete concept. That is, a respondent was said to be cooperative if in a particular query situation she performed a particular cooperative act (such as correcting a misconception or over-answering a question), and was considered uncooperative if she failed to perform that act in that situation.

But this is an oversimplification that is reasonable only in the restricted kinds of query situations treated by questioner-based models. First of all, a respondent not only may choose *whether* to cooperate with a questioner at all but also *to what degree* she will cooperate. This fact is apparent to anybody who has ever dealt with a government agency or any so-called "customer service" department. A respondent's willingness to cooperate—the degree to which she intends to cooperate—may affect:

- the amount of resources she chooses to apply in detecting the conditions that motivate cooperative goals. For example, a greater willingness to cooperate may lead a respondent to devote more resources to the task of identifying the questioner's goals, plans, and needs. This may in turn uncover more conditions requiring attention in the response;
- her decisions regarding what cooperative response goals to adopt. For example, a UNIX consultant might respond to a query like "What does the df command do?" by referring the questioner to the UNIX manual rather than actually answering the question;
- how much effort the respondent expends trying to achieve her cooperative response goals. For example, correcting a misconception could be achieved with a simple denial, or with a more elaborate response (like the ones that ROMPER produces).

Furthermore, a respondent's willingness to cooperate appears to be not an independent variable, but a product of other variables, such as the respondent's attitude toward the questioner (e.g., a person is likely to be far more willing to cooperate with her spouse than with a stranger encountered on the street), the amount of time available for the interaction, and so on.

One might argue that factors such as willingness to cooperate and its attendant variables can be ignored in computer models of discourse without loss of functionality (although see [Hovy 88] for some discussion of how interpersonal considerations might affect language generation), suggesting instead that question-answering systems be designed to be "maximally cooperative", or "as cooperative as possible". But this begs the question, because determining the amount of cooperativeness (assuming that such a thing can actually be quantified) that is desired or appropriate in a given situation appears to be part of the very task of cooperative reasoning itself. Moreover, systems cannot in general get around this problem by saying everything they can think of. As Grice observed via his Maxim of QUANTITY [Grice 75], speakers who are perceived as saying too much may appear to be uncooperative, or worse, their loquacity may lead to unwanted implicatures.

My point is that the concept of "willingness to cooperate" makes sense only in a respondent-based framework, that is, in a theory defining what it means to be cooperative. Because questioner-based models simulate fixed situations, the respondent's decision regarding how cooperative to be is essentially "hard-wired", non-varying, and thus factored out of the model.

### Minimal requirements

Although respondents appear to have some flexibility in deciding how cooperative they want to be in a given situation, this mostly seems to involve choosing how far to exceed the minimal requirements of cooperation demanded by the situation. That is, in any given question-answering situation, there seems to be a sense that R has a set of minimal obligations to be met in her response. Failing to satisfy any of these would cause her to be perceived by Q as uncooperative. Consider the following example from the UNIX transcripts:

(38) Q: For a while now, I have been trying to use Unix by printing out a lot of "man" files. These, however, are unwieldy and hard to reference. Is there a book from which these are taken, and if so, where can I get it?

R: You can purchase the UNIX manual in its entirety in the Comp Center library on the second floor of Evans. This will set you back 30+ bucks and will not be accurate in all cases. This is because CF&O doesn't like to change to new operating systems, while around here, we boot the latest version and hope that it works. If you're really looking for the last word, get friendly with someone who has the 4.2 version manual.

Note that R's response might very well have been perfectly cooperative and acceptable to Q if she had said no more than, "You can purchase the UNIX manual in its entirety in the Comp Center library on the second floor of Evans". I would argue that in the conversation above, R had a minimal obligation to address Q's desire to obtain a hardcopy of the UNIX manual. In the research framework I outlined earlier, this might suggest that some cooperative goals are obligatory while others are optional. If R fails to achieve her obligatory cooperative goals (or at least address the conditions that motivate them), she will appear uncooperative. The information that R adds above and beyond the minimal requirements reflects a decision she has made to expend more effort on Q's behalf than the situation requires—a decision that might follow from other goals that R has, such as improving her relationship with Q (if Q held some position superior to R), impressing Q with her helpfulness, causing Q to recommend her for a better job than as a UNIX consultant, and so forth.

As a corroborating factor, apologies, explanations, or justifications seem to be required in a conversation when a respondent is unable to meet her obligations. For example, if R did not know where Q could buy a copy of the manual, she might be obliged to respond with something like:

(39) R: I know there is such a manual, but I'm afraid I don't know where you could buy it. Have you tried the bookstore?

Here, in order to make it clear that she is still trying to be cooperative despite her inability to help Q obtain a copy of the manual, R implicitly apologizes for her lack of knowledge and goes on to suggest an alternate course of action for Q to take. The point is that while a respondent-based theory of CRG can and should account for a respondent's degree of intention to cooperate, it should also account for R's understanding of her minimal obligations to a questioner in a conversation, and for how her inability to meet any or all of those obligations might affect her response behavior.

### The limits of knowledge

Rather than being an absolute, cooperativeness varies with the quantity, quality, and degree of detail of a respondent's knowledge. This is not an issue in simple domains, but it becomes noticeable in the richer domains that realistic natural-language systems will eventually have to deal with. Consider the following example from the UNIX transcripts.

(40) Q: How do you change mode for a dotted file (such as .login)?
R: You would use the same method as for any other file you own. Basically, just type "chmod mode filenames" where mode is "a*b" with "a" equal to any combination of "u" (user), "g" (group), and "o" (others). "*" replaced by "+" (add permission), "-" (remove permission), or "=" (set to specified permission). Finally, "b" is replaced by a combination of "r" (read), "w" (write), and "x" (execute). Read the manual page if you would like more explanation of exactly what everything means.

The response seems to be a perfectly reasonable and cooperative one. Q appears to hold a misconception that there is a special way to change the access mode of dotted files, a way different from that for normal (non-dotted) files. R has detected that apparent misconception, since she begins her response by explaining that there is no such special method. But given that Q raised the question, it seems perfectly reasonable for R to suppose either that Q does not know how to change the mode of *any* file, or does not know the correct command syntax. Consequently, R goes on to carefully describe the use of the `chmod` command.

For the CRG theorist, one of the most revealing features of the UNIX transcripts is that they occasionally contain more than one response to the same question.[8] Such is the case with this example. A second respondent replies as shown in (41).

(41) R: The problem might be that you don't own your .login file. To save space, most of the .login files are linked together (i.e., they're all one file). To fix this, type the following:
cp .login .login.copy
rm .login (or rm -f .login, if that doesn't work)
mv .login.copy .login
Then you can chmod it, edit it, or whatever.

Evidently the second respondent has a piece of knowledge that the first lacks, namely, that `.login` files on the particular computer system used by Q are typically linked together.

[8]Since the users' queries were mailed to a distribution list, occasionally more than one consultant would choose to reply. Sometimes the different responses simply reflect the different views held by the various respondents. Other times, it appears that one response is meant to clarify, augment, or correct one of the other responses (since a copy of each consultant's response was sent to the distribution list, so all consultants saw all responses to all queries).

This has certain consequences, one of which is that the file will not be owned by Q, which in turn prevents him from (among other things) editing it or changing its access mode.

This special knowledge enables some deep reasoning on R's part. First, she can comprehend the significance of Q's use of `.login` as a particular example. Hypothesizing that Q's question might be motivated by some difficulty he is having operating on the `.login` file, R is able to identify the likely source of Q's problem—that he does not own his `.login` file—and then produce the more helpful response of the two. This is very much in accord with the "explanation-based" reasoning approach advocated by Quilici *et al.* (see Section 2.3.2).

Obviously, the respondent's chain of inference in the second example is subtle and is enabled only by her possession of a piece of domain-specific knowledge. This suggests that the amount of knowledge a respondent possesses will affect her ability to detect the conditions that motivate cooperative goals. In the first case, R's relative lack of knowledge only permits her to conclude that Q has a misconception. In the second case, R's richer knowledge allows her to construct a detailed representation of Q's predicament and thus recognize that different cooperative goals (other than misconception correction) are well motivated.

Another significant point is that both responses are cooperative. We certainly do not want to fault the first respondent for not knowing an arcane policy detail. But even in the absence of that knowledge, she can still be cooperative, although her response may perhaps not be as helpful as Q needs or would like. On the other hand, if the second respondent, in full possession of her unique knowledge, had provided the same response as the first, we would be inclined, I think, to fault her for being uncooperative. This suggests that any definition of cooperative behavior must be established relative to the respondent's knowledge, that is, we need a notion of a respondent's acting cooperatively "within the limits of her knowledge".

This pair of examples argues strongly that it is wrongheaded to define cooperativeness in terms of the kinds of acts that respondents perform in various query situations. Furthermore, it suggests that it will not be possible or useful to formulate an objective list of criteria according to which a response can be judged "cooperative". Instead, we need to model cooperative behavior as I have argued: in terms of sets of goals that respondents try to achieve in their responses.

### Evaluating goals

A respondent's experiences and opinions influence how she chooses to cooperatively respond to questions. Let's consider another interesting pair of responses taken from the UNIX transcripts. We have already seen the first of the pair in example (38). In this example, Q has indicated a goal of obtaining a hardcopy of the UNIX manual pages. R seems to consider this a perfectly reasonable goal and so she tells Q where to get the manual, including other pertinent details such as its price and accuracy. In fact, what she actually does above and beyond giving Q the information he needs to achieve his goal is *critique his plan.* That is, Q has indicated he has a general plan for operating in the UNIX environment: when he gets stuck, he wants to be able to refer to a hardcopy of the manual. But R has evaluated this plan and found it to have some shortcomings: first, the manual is costly, and second, the information it contains will not always be accurate. R exceeds her minimal response obligations because she believes that Q might find her comments on his plan useful.

Now compare the first response with one from a different consultant, as shown in (42).

(42) R: I think you'll find that having a hardcopy of the manual might not be of too much help. If you are just starting out with UNIX, you might want to get "Unix for Beginners" or some other intro doc from the CF&O library (2nd floor). They have a list of the publications they sell in the library. You might also wish to purchase an intro doc to vi, the text editor of choice around here.

In this second case, R's beliefs about the utility and reasonableness of Q's plan (to refer to the UNIX manual when he gets stuck) with respect to his more general goal (of learning how to use UNIX and/or of accomplishing tasks in UNIX effectively) suggest to her that there are better plans than the one Q has proposed. Her ability to evaluate plans in this way is one of her resources as a cooperative respondent—one of the ways she identifies conditions that motivate cooperative response goals.

Once again, we would neither want to declare one response cooperative and the other not, nor want to rank one response "more cooperative" than the other. Rather, the examples suggest that in fulfilling their cooperative conversational goals, respondent-based dialogue systems may need to be able to question and take issue with the plans and goals proposed by questioners and choose responses accordingly.

### Mode and communicative effort

The *mode of communication* affects how the response to a given query is formulated. Consider the distinction between what might be called *real-time* and *delayed-response* interactions. The dialogues recorded in the UNIX transcripts are all examples of delayed-response interactions: the questioner formulates his query and transmits it to the consultant, who reads and responds to it at some time in the future. There is a similar delay for the consultant, as her response also will be read at some unknown time after it is sent. In interactions of this sort, we often see questions presented in detail, as the questioner appears to be including all the information that he or she believes could possibly be helpful to or needed by the consultant in formulating a reply. The questioner's goal seems to be to avoid lengthy exchanges where the consultant must request further information or ask for clarification. A typical example is shown in (43).

(43) Q: What is "east"? Whenever I am logged on through a 1200 baud modem and do a "w" command the machine says that I am doing "east" but when I type "east" I am asked for a login name and password. It is apparently a host of some sort but why am I logged on through it when I call Cory?

The questioner could simply have asked, "What does 'east' do?" but that might not be enough for R to be able to provide a useful reply, or the kind of reply that Q is interested in. But rather than take the chance that R might find it necessary to ask for clarification, Q transmits all the details he believes R might need, both to understand the question and to produce a relevant response. Responses are often equally elaborate, for the same reason: the respondent wants to avoid putting the questioner in the position of having to request further information. This suggests that the respondent's knowledge of the cost associated with each turn of the conversation may influence which goals, and perhaps how many of them, she decides to attempt in a single response.

In contrast, real-time interactions (like those recorded in the EMACS transcripts) have the opposite characteristics. Since questions can be asked and answered relatively quickly, the cost of providing detailed, carefully explained and justified answers may be considerably greater than the cost of being concise and allowing one's interlocutor to make specific requests for elaboration and clarification. Thus dialogue participants in real-time interactions tend to communicate in terse phrases and make more liberal assumptions about each other's knowledge, of both the world and the state of the conversation. This is particularly evident in the collaborative-task dialogues recorded in the EMACS transcripts; a typical example is shown in (44).

(44) Q: Beginning of region. How?
R: Type ESC-m. It should say mark set at the bottom of the screen.

Q seems able to draw upon his beliefs about what is mutually believed about the state of the conversation when framing his query (which otherwise is nearly uninterpretable).

Of course, the abbreviated nature of real-time interactions tends to increase the chance of miscommunication and the need to enter "debugging dialogues", as the example in (45) illustrates.

(45) R: Yes. At what point are you being kept?
Q: I am at the beginning of the text to be edited.
R: I mean, how much of the macro defn, [sic] have you finished?
Q: I entered the search command and then closed the macro.

In a delayed-response interaction, R would have had the time to plan a clearer, more specific request for information.

Apart from the mode, the pure physical *effort* involved in carrying out acts of communication seems to affect how respondents formulate their replies. For example, rather than typing a description of how a particular UNIX command works, a consultant might simply refer the questioner to the appropriate manual page, as shown below.

(46) Q: How do the 'history' and 'alias' commands work? There do not seem to be 'man' entries for them.
R: Try "man csh".

But if the communication medium permitted speech, the respondent might be more willing to provide a verbal explanation. This seems to tie in with the issue of how much responsibility the respondent has to address the needs of the questioner in a given dialogue setting.

The point of this discussion is that both the mode of communication and the effort involved in communicating are variables that affect cooperative behavior. But these variables have no place in questioner-based models, since such models factor out the process by which a respondent chooses her method of response, the very process that would be influenced by those variables.

#### Respondent's goals

As shown in UCEGO [Chin 88a], if a respondent is allowed to have her own goals in the world, separate from those of the questioner, then there is a possibility, if not a likelihood, that they will have an important influence upon how replies are chosen. Should her goals conflict with those she believes are held by the questioner, she might decide to be uncooperative or perhaps try to persuade the questioner to adopt different goals. On the other hand, if her goals complement those of the questioner, she might be motivated (subject to her sense of responsibility and willingness to cooperate) to modify her response (and perhaps her own goals) accordingly. For example, if Q were to ask R for directions to some particular place and it so happened that R was going there herself (or even just in that direction), rather than giving the directions she might simply invite Q to accompany her.

## 3.3 Concluding Remarks

In this chapter I have characterized two perspectives on cooperative response generation, distinguished on the basis of whether the point of view from which the principles of cooperation are defined is the questioner's or the respondent's.

#### The questioner-based perspective

I characterized the questioner-based perspective as defining the principles of cooperative reasoning from the questioner's point of view, devising methods that natural-language dialogue systems can use to perform certain predefined kinds of "cooperative" behavior. Development of questioner-based models has followed a three-part research program:

1. informally identifying the needs, desires, preferences, and expectations of questioners that are potentially relevant to the processing of a query in a given query situation;

2. identifying response strategies that can be used to meet the needs, satisfy the desires, accommodate the preferences, and live up to the expectations of a questioner in a given query situation;

3. developing representations of query situations and designing reasoning procedures on those models that can enable dialogue systems in the modeled situations to respond using the desired response strategy.

I then claimed that the systems resulting from this research program have attempted to achieve goals of one of the following three general types:

1. avoiding adverse effects on the questioner;

2. improving a natural-language interface's ease of use;

3. adding useful functionality to a question-answering system.

I argued that in no case were the specific cooperative goals that a given system was designed to achieve ever formally stated, and that as a result the reasons for adopting those goals were never defined.

In exploring the computational consequences of the questioner-based approach, I developed an abstract architectural model of previously-developed CRG systems. Using this architecture, I argued that the questioner-based approach has created an army of models of restricted question-answering situations in which predefined responses were generated and declared to be "cooperative". Because the approach permits the situation models to be defined inconsistently across theories, the resulting theories have turned out to be largely incompatible with one another.

I claimed that the theoretical consequence of the questioner-based approach is that the resulting models have only *described* response behavior that seems to be cooperative. The models have given no guidance regarding the principles that *drive* cooperative behavior.

Finally, I argued that more general cooperative dialogue systems will have to *integrate* different forms of cooperative behavior in their responses. But questioner-based models have been developed in different domains with different implicit assumptions, have considered different aspects of the response-generation process, and have focused on developing machinery to generate particular responses rather than on identifying the general reasoning principles from which those responses are derived. These problems stand in the way of integration and so I conclude that the questioner-based approach is ultimately the wrong one to take if we want to develop truly cooperative natural-language dialogue systems.

### The respondent-based perspective

In opposition to the questioner-based perspective, I described the respondent-based perspective as identifying and answering questions concerning why and how respondents choose the goals that give rise to behavior that is perceived as cooperative. At the heart of the respondent-based perspective is the view that *responses reflect the respondent's goals*—i.e., that the natural-language utterances of a response should be treated as realizing, in concert, a plan the respondent was using to achieve certain goals.

Taking that view as a hypothesis, I distinguished five different elements of a respondent-based model of cooperative response generation:

1. the goals that respondents adopt and try to achieve in their responses;

2. the conditions of the conversation that motivate various response goals;

3. the knowledge and reasoning resources needed to determine the status of the goal-motivating conditions;

4. the reasoning processes and principles by which response goals are actually adopted;

5. the strategies that may be used to achieve a given response goal.

My claim is that a consistent program of research aimed at fleshing out the details of these five theoretical elements will help us both to build more capable cooperative dialogue systems and to better comprehend the nature of cooperative behavior. In the rest of this dissertation, I will develop a theoretical framework within which we can explore problems in CRG from the respondent's perspective.

# Chapter 4

# Cooperative Response Planning Systems

In Section 3.2, I introduced a new perspective—the *respondent-based perspective*—on cooperative response generation and argued that it should be used as the theoretical foundation for the design of natural-language dialogue systems that are to function as cooperative respondents rather than serve as models of particular kinds of cooperative responses. At the heart of this new perspective is the language-as-action principle that responses reflect the respondent's attempt to achieve one or more high-level response goals. But if responses are to be viewed as actions planned to achieve goals, it follows that cooperative respondents should be viewed as *planning agents.*

That conclusion sets the stage for the rest of this dissertation. My goal is to characterize the design principles and general structure of a cooperative response generation (CRG) system whose core component is an agent capable of forming and adopting goals and developing plans to satisfy them. Although this is only one of its components, I will nevertheless refer to the overall system as a *Cooperative Response Planning System*, or CRPS.

In this chapter I will discuss a proposed architecture design for a CRPS. First I will consider its top-level structure and then examine the architectures of its critical processing elements, paying particular attention to the organization of the planning component. Although my long-term goal is to identify the kinds of knowledge and reasoning processes needed to generate responses of the sorts found in naturally occurring help dialogues such as the UNIX and EMACS transcripts, the particular CRPS design that I will be describing here is to be seen only as a first approximation to a theory of cooperative response planning.

The principal contribution of the architecture is its decomposition of the CRG problem into subtasks with limited channels of communication. I will argue that the divisions I impose are well motivated from naturally occurring data. Furthermore, I will show how the design illuminates theoretically-significant problems that from a questioner-based perspective would be overlooked. Identifying these problems is an important step toward the development of "truly cooperative" natural-language dialogue systems. Although much work remains to be done before they can be solved, I believe that simply posing them will provide insight into the complexities of CRG when the task is approached from the respondent's viewpoint.

I will begin with a description of the overall architecture of a CRPS. Next, I will focus on the design of its first major processing element, the module that constructs the dynamic knowledge base from which the planner reasons. I then take up the examination of the planner and sketch its operation.

## 4.1 Top-Level Architecture of a CRPS

I distinguish three functional reasoning components comprising the top-level architecture of a CRPS: the *Conversation Modeler*, the *Response Planner*, and the *Language Generator*. The corresponding architecture diagram is shown in Figure 4.1. The arrows indicate flow of data and control between the components. Here I will describe them briefly.

**Conversation Modeler** An important concern in the design of any planning system is the question of where its goals originate. A CRPS is different from other planning systems in that it does not acquire its goals from some external source but rather *forms them itself*. This is because choosing the goals that the system will pursue is itself one of the crucial stages of the process of cooperative response generation.

The goals that a CRPS forms and attempts to achieve are motivated at least in part by its interpretation of the questioner's input utterances and by its understanding of the purpose and direction of the conversation in which it is participating (cf. Grice's statement of the Cooperative Principle, discussed in Section 2.1). This implies that a CRPS needs to build and maintain some kind of model of the ongoing conversation. It is in this *conversation model* that the initial stimulus for the system's response-directed reasoning will be found. The conversation model is a dynamically-constructed augmentation of the planner's basic knowledge base, and is built by the Conversation Modeler. The conversation model's general role in cooperative response planning will be discussed in more depth in Section 4.2.

**Response Planner** By reasoning from the conversation model, the Response Planner both forms the goals it will attempt to achieve in the system's response and develops a corresponding response plan. During its operation, the Response Planner may extend the conversation model with data derived from its attempts to form, adopt, and satisfy goals. The response is emitted as a collection of *surface goals*—goals specified in sufficient detail that they can be achieved directly using natural-language expressions. The Response Planner is the center of the system's cooperative response reasoning; the principles and general structure of its design will be considered in Section 4.3.

**Language Generator** The Language Generator is the linguistic planning component of the CRPS. It takes the surface goals generated by the Response Planner and constructs natural-language text to realize them. While interesting, the design and operation of the Language Generator and the details of its interface with the Response Planner are concerns that are beyond the scope of the present enterprise. Instead, I will make the gross simplifying assumption that the surface goals emitted by the Response Planner can be

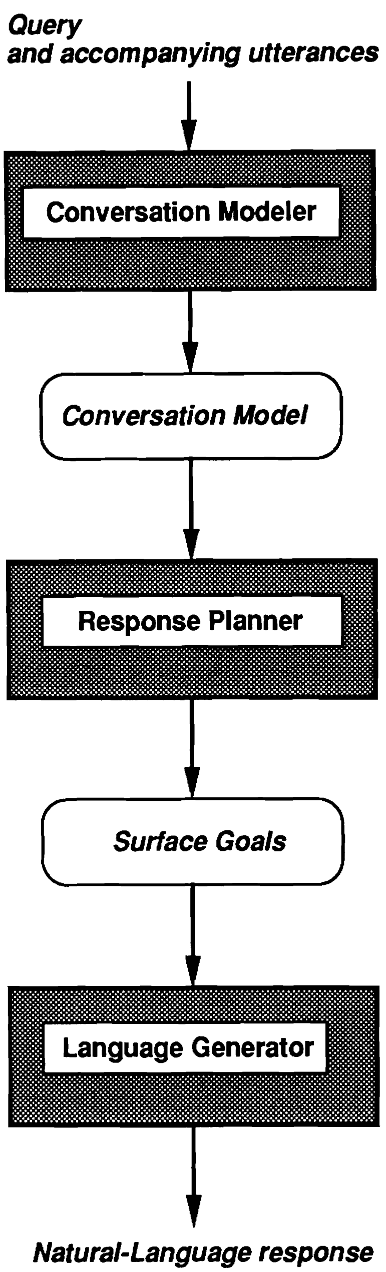


Figure 4.1: Top-Level Architecture of a Cooperative Response Planning System

sufficiently well specified for the Language Generator to realize them directly using expressions in the target language. For more detailed exploration of some of the complexities of the natural-language generation process, see (among many others) Appelt [Appelt 85b], Hovy [Hovy 88], Moore and Paris [Moore 89], and Rubinoff [Rubinoff 90].

## 4.2 The Conversation Modeler

In domains of realistic complexity, natural-language dialogue systems will necessarily have to maintain general knowledge bases of substantial scope and detail. The beliefs they hold will be of many different kinds, including but not limited to:

- beliefs about the kinds of properties objects in the world may have;
- beliefs about the current (and possibly past) values of those properties;
- beliefs about the static or dynamic nature of those properties;
- beliefs about general characteristics of the world;
- beliefs about possible ways in which the world changes over time;
- beliefs about the ways in which actions in the world can alter states of the world;
- beliefs about other agents that operate in the world;
- beliefs about the system itself;
- beliefs about the ways in which linguistic actions can be used to affect the mental states of other agents.

How a CRPS acquires its general knowledge base is not of concern here.

During the course of a conversation, a CRPS constructs a special database of beliefs that are relevant to the dialogue. This database, which I call the *conversation model*, contains at least the system's interpretation of the questioner's utterances. However, I will argue that other reasoning resources may need to be invoked—augmenting the conversation model—before the Response Planner has enough data from which to construct an adequate reply. (The conversation model may also be extended during the Response Planner's operation; some ways in which this occurs are discussed in Section 4.3.6.)

In the architecture shown in Figure 4.1, the *Conversation Modeler* is the component that constructs the initial conversation model. It processes the natural-language input received from the questioner and builds a structure that represents those details of the question-answering context, derived both directly and indirectly from the input, that are relevant to the system's selection of a cooperative response. Note that the Conversation Modeler does not provide goals for the system; rather, it provides the data from which the Response Planner will reason to choose response goals.

### 4.2.1 The conversation model

Before a CRPS can begin to choose an appropriate and useful response, it must start with some kind of representation of the query that was posed to it. But a representation of the query alone is not in general sufficient to support the range of response goals that cooperative respondents appear to adopt. If a CRPS is to demonstrate the same flexible range of abilities that human cooperative respondents have, it must construct a richer, more detailed model representing not only its beliefs about the questioner that can be derived directly from his natural-language utterances, but also those beliefs whose derivation requires the application of special reasoning procedures such as plan recognition and explanation generation.[1]

I call this model the *conversation model* because it represents the relevant details of the "current conversation". However, despite the use of the word "conversation", I will not be considering the complexities of CRG that are a function of extended discourse. Rather, I will view the conversation model as derived from the processing of a single (possibly multi-sentential) input "turn" in isolation from previous discourse. Extending the model to cope with both discourse and mixed-initiative dialogue (such as clarification subdialogues [Litman 84b]) is a separable research problem.

The conversation model is a kind of "relevant knowledge pool"—a dynamically-constructed knowledge base containing all the information made explicit in or inferred from the questioner's utterances that is initially useful to the response-selection process. *It is a set of beliefs, derived from the system's general knowledge store, which may motivate the system's initial response goals.* (The use of the word *initially* here is significant, as the planner may change its goals during its operation—see the discussions in Sections 4.3.5 and 4.3.6.)

Identifying all the elements that must be part of the conversation model (and developing adequate representation schemes for them) in order to support cooperative response generation remains an important ongoing research task. Among the reasonable candidates for inclusion are these:

- the questioner's query;
- intensional representations of entities referred to (as described in the questioner's utterances);
- evaluations of the well-formedness of referring expressions;
- referents, if they exist, of referring expressions and parts thereof;
- beliefs about the questioner's goals, plans, preferences, and constraints;
- other beliefs about the questioner's beliefs and intentions derived through explanation-based reasoning.

This of course is not an exhaustive list. The point is to suggest that in realistic NLQA applications, the demands of cooperative response generation may require a conversation model that is quite rich indeed.

[1] I am using "explanation generation" here in the same diagnostic sense as in Quilici *et al.* [Quilici 88], that is, a process of generating a plausible explanation (represented as a chain of beliefs) for why the questioner might hold a particular valid or invalid belief.

### 4.2.2 Theoretical significance of the conversation model

In separating "conversation modeling" from the rest of cooperative response planning, I am claiming that we can identify some set of knowledge-bearing reasoning resources that are routinely invoked prior to response planning. That is, there is a distinct stage of cooperative response planning during which the system amasses a body of information which it then uses as a basis for forming its response goals. The richness of its conversation models critically affects a CRPS's ability to provide useful and helpful responses; consequently, developing a theory of the conversation model involves making claims regarding the knowledge that cooperative respondents must have available or be able to derive in order to function properly.

Previous research efforts in CRG, in the course of modeling different manifestations of cooperative response behavior, have developed limited theories about the knowledge structures that NLQA systems might require to produce particular forms of behavior under study. Certain structures have been proposed as part of the respondent's general domain knowledge (for example, Mays's temporal axioms [Mays 84b], McCoy's perspective-enhanced domain model [McCoy 88], and the plan libraries that are required by most models of plan inference), while others have been proposed as derived from or updated during a dialogue (for example, Carberry's context model [Carberry 88], Pollack's explanatory plans [Pollack 86], and the explanation chains constructed by Quilici *et al.*'s AQUA system [Quilici 88]).

But when one tries to envisage a system that will possess a repertoire of cooperative abilities, questions about both the *theoretical status* of and the *relationships* among all these knowledge structures become significant. For example, one might ask of a knowledge structure or knowledge-bearing procedure whether it represents an actual claim about the cognitive capacities of cooperative respondents or is instead a tool designed only to make a particular response simulation work. Of particular interest is the question of how different structure-building operations participate in the response-planning process. Which ones are performed routinely prior to response planning and which are invoked only when demanded by the response planner? These are challenging questions that arise only when the CRG problem is approached from the respondent's perspective.

The Conversation Modeler is conceived of as comprising only those structure-building resources that are routinely invoked prior to response planning. This gives rise to the following question: how can we determine whether a given reasoning procedure is routinely performed prior to response planning? I do not have a satisfactory answer to that question; for the present, I offer the following pair of conditions as a first approximation to an answer:

**(C1)** The procedure must produce information necessary for a CRPS to identify the range of goals it needs to satisfy in order to produce responses that are adequately cooperative.

**(C2)** There is no reasonable way to decide invoke the procedure on the basis of other evidence that could be found in the conversation model.

To grasp the intuition behind these two conditions, let us consider the process of goal inference. Research has shown that when deciding how to reply, respondents seem to require access to beliefs about the goal or goals that underlie a questioner's utterances (see, for example, [Allen 83]). Examples (47) and (48) demonstrate that respondents will often request more information if they believe that their comprehension of the questioner's goal (and plan) is inadequate for selecting a helpful reply.

(47) Q: What are the rules concerning use of TELNET? Can I use it when the system is lightly loaded, or not at all. Please advise me.
R: The use of telnet to reach remote arpa sites is restricted to official arpa business only. What particular site to do you want to get to?

(48) Q: How do I erase the binding?
R: You don't, but why would you want to? (You ought to always choose ctrl-H or some unused key to bind it to in the first place of course).

In (47), R can infer that Q wants to connect (using TELNET) to some remote computer. Her response can be interpreted as correcting an evident misconception that permission to use TELNET depends upon system load instead of upon the purposes for which the network is being used. This correction, however, does not address Q's desire to connect to a remote computer. But in order to address that goal, R needs a specification of the machine to which Q wishes to connect (so that she might determine whether it is reachable by means other than TELNET). That is, R needs a more precise understanding of Q's goal in order to provide an adequately cooperative response.

In (48), it seems quite clear that R is attempting to ascertain the goal that Q aims to achieve by "erasing the binding". Once again, R seems to be motivated by a goal of ensuring that her response has left Q in a position to achieve his goal (or some reasonable substitute therefor). These two examples illustrate that respondents reason both from and about their beliefs about questioners' intentions.

Thus the ability of agents to respond cooperatively seems to depend upon the availability of knowledge about questioners' goals. Except in artificially-constrained domains, an agent who lacks (and cannot acquire) such knowledge will often find herself unable to fulfill her responsibilities as a cooperative respondent. More importantly, as yet no procedures are known either for deciding whether the results of goal inference might be needed in order to produce an adequately cooperative response or for computing how many of Q's goals and the relationships between them R needs to infer. (This is because questioner-based models never needed to make such decisions.) Goal inference simply seems to be one of the reasoning procedures that respondents must at least attempt before they have a sufficiently solid foundation for deciding how they ought to respond.

The two conditions stated above try to capture the intuition that there is some minimum set of reasoning procedures that must be invoked to build a sufficiently solid knowledge foundation for a cooperative respondent's response-planning apparatus. Condition C2 is intended to restrict that set to just those procedures whose invocation cannot be made conditional on the results of tests on the output of other elements of the Conversation Modeler's collection.

To see how Condition C2 might justify excluding a procedure from the Conversation Modeler's collection, let us consider the addition of a *plan synthesis* procedure. We could expect a plan synthesizer (such as Luria's KIP planner [Luria 88]) to be called when the cooperative response planner decides that it needs to construct a new plan for a goal. The need for such a procedure is suggested by examples such as (49):

(49) Q: For a while now, I have been trying to use Unix by printing out a lot of "man" files. These, however, are unwieldy and hard to reference. Is there a book from which these are taken, and if so, where can I get it?

R: I think you'll find that having a hardcopy of the manual might not be of too much help. If you are just starting out with UNIX, you might want to get "Unix for Beginners" or some other intro doc from the CF&O library (2nd floor). They have a list of the publications they sell in the library. You might also wish to purchase an intro doc to vi, the text editor of choice around here.

We might reasonably suppose Q's top-level goal to be something like, "Q is able to perform tasks in UNIX," and that Q believes that "having a hardcopy of the UNIX manual" is a step toward that goal. Given such an analysis, R appears to question the usefulness of that step, suggesting what she considers to be a more advantageous approach. In order to offer such advice, R presumably must be able to synthesize a plan that from her point of view is preferable to the one inferable from Q's utterances. But does this imply that a plan synthesizer should be incorporated into the Conversation Modeler?

Not necessarily. Consider the response provided by a different consultant, shown in (50).

(50) R: You can purchase the UNIX manual in its entirety in the Comp Center library on the second floor of Evans. This will set you back 30+ bucks and will not be accurate in all cases. This is because CF&O doesn't like to change to new operating systems, while around here, we boot the latest version and hope that it works. If you're really looking for the last word, get friendly with someone who has the 4.2 version manual.

Ignoring the last sentence, this response gives no indication that R performed any plan synthesis at all. (The last sentence seems to demand an explicit decision made later to invoke a planner.) That in itself is not sufficient to discount plan synthesis as a Conversation Modeler operation. However, there also seem to be reasonable bases for deciding whether or not to invoke a plan synthesis mechanism. For example, the respondent might be able to evaluate the questioner's plan according to such metrics as simplicity or conventionality without resorting to the construction of a plan of her own. If that were possible, the response planner would be able to call the plan synthesizer on demand rather than routinely. Indeed, the decision to call a plan synthesizer might be heavily influenced by the degree to which the respondent is willing to commit resources on behalf of the questioner: if the questioner's plan looks workable, the respondent might simply decide not to make the effort to determine whether a better plan exists.

The point of this discussion, however, is not to argue against the routine invocation of a plan synthesizer when building the conversation model; whether it is or is not a routine computation is an open question. Rather, I am suggesting that conditions C1 and C2 work in concert to provide at least a rough guideline for distinguishing the knowledge-bearing resources that function as part of a respondent's basic apparatus for perceiving and understanding language from those resources over whose application the respondent seems to exert conscious control.

### 4.2.3 A simple version of the Conversation Modeler

The discussion so far has examined the general role and theoretical status of the Conversation Modeler in a cooperative response planning system. In this section I will define a simple version of the Conversation Modeler. This should help to clarify some of the issues with which designers of response planning systems need to be concerned.

This limited version of a Conversation Modeler assumes the following subset of knowledge elements mentioned earlier:

- a representation of the questioner's query;
- a representation of the most general goal relevant to the conversation[2] that the respondent believes the questioner is trying to achieve (hereinafter referred to as the *top-level goal*);
- a representation of the questioner's plan that relates the query to that goal;
- a set of comments representing the respondent's *evaluation* of the questioner's plan.

I conjecture that conversation models must be *at least* this rich in order to support CRG.

A representation of the query is needed if only as a clue to the information that Q believes he needs. Without access to some model of the query, the respondent would be unable to provide an answer if she believed it would be useful. Strong arguments supporting the need for knowledge of both Q's plan and goal have been made by many others: such knowledge is the key to modeling responses that address the purposes underlying Q's query rather than just the query itself. Finally, Pollack convincingly argues [Pollack 86] that inferred plans must be able to reflect questioners' underlying incorrect beliefs; consequently the conversation model must also contain annotations indicating the respondent's evaluation of the plan, noting the kind (and perhaps also the source) of any errors detected. The architecture of a Conversation Modeler capable of producing these elements is shown in Figure 4.2.

As shown in the figure, the Conversation Modeler consists of three sequential processing elements: an *input analyzer*, a *plan inference mechanism*, and a *plan evaluation mechanism.*

#### Input analyzer

The input analyzer translates the input natural-language utterances into a suitable internal representation language. The design of this component is beyond the scope of this effort; an adequate design will be assumed.

[2]The notion of the most general goal *relevant to the conversation* is a rather subtle one. Exactly how it might be mechanically identified is an open research problem. To the best of my knowledge, all current models of plan inference never have to determine the top-level goal; it is always provided in the input or otherwise made explicit. Pollack's SPIRIT system, for example, is always given the questioner's top-level goal [Pollack 86], and Allen's inference system works with a two-element space of possible top-level goals [Allen 83].

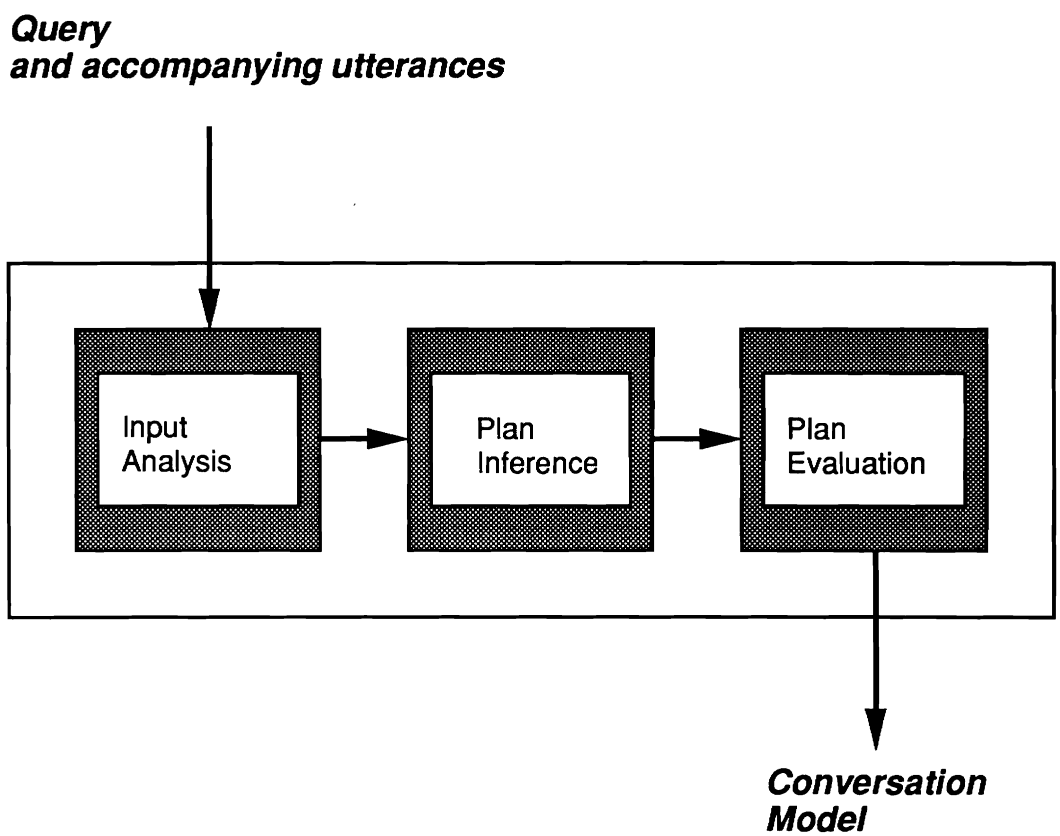


Figure 4.2: Architecture of the Conversation Modeler

### Plan inference mechanism

The plan inference mechanism takes as its input the data generated by the input analyzer and constructs a model of the plan that the system believes the questioner is pursuing, based on its interpretation of his utterances.

I will make the assumption that from the questioner's input utterances, the Conversation Modeler can identify both a *request for information* and a *top-level goal.* The request represents the information literally requested[3] by the questioner. The top-level goal is the

[3] Although I will not consider the processing of *indirect speech acts* [Searle 75] here, it is my position that queries do have context-independent, literal interpretations that must be retained somewhere in the system's model of the conversation (but cf. [Allen 89, Hinkelman 89, Hinkelman 90] for a different point of view). While respondents may ignore the literal interpretation of an indirect request when framing a reply, this reflects a *choice* that they have made. Moreover, a CRPS cannot be denied access to a query's literal interpretation because it may need to respond to the literal request if the indirect request cannot be satisfied, as shown in (51).

(51) Q: Can you tell me the time?

most general goal the questioner is pursuing that is relevant to the current dialogue.

I will assume that the plan inference mechanism can infer an unambiguous plan (cf. my discussion and analysis of the work of van Beek and Cohen on page 40). This does not seem to be a serious limitation, and it eliminates the complexities of reasoning involved in deciding what to do when the system cannot determine exactly how the questioner plans to achieve his top-level goal.

Like Pollack, I will not make the *valid-plan assumption* [Pollack 86]—the assumption that the questioner's plan represents a correct way to accomplish his goal—because recognizing and addressing the mistakes people make in forming their plans is an essential part of cooperative behavior.

#### Plan evaluation mechanism

Not only am I following Pollack in abandoning the valid-plan assumption, I am also following her in making a plan evaluation procedure a required step in the process of building conversation models. The plan evaluation mechanism (or simply, the plan evaluator) takes as input the plan structure produced by the plan inference mechanism. It inspects that structure and annotates it in a variety of ways, indicating any and all errors it finds and their consequences. Interesting research problems include designing a plan evaluator (and especially exploring the relationship between plan inference and evaluation) and determining a useful set of plan annotations.

### 4.2.4 Consequences of incompleteness and incorrectness

With respect to the beliefs comprising the planning agent's reasoning environment—both those held at the initiation of the dialogue and those that are derived from the questioner's utterances to form the conversation model—at least two assumptions are possible: *completeness* and *correctness*. The completeness assumption implies that the agent possesses exhaustive knowledge of her environment. By that assumption, if in the "real world" David Dinkins is the Mayor of New York City, the agent must also have that as one of her beliefs. The correctness assumption, on the other hand, implies that all of the agent's beliefs are "correct", that is, in accord with some notion of what might be called "objective truth". Thus if the agent holds a belief that "David Dinkins is the Mayor of New York City", and the agent's beliefs are assumed to be correct, then in the "real world" the person denoted in the agent's belief space by "David Dinkins" must actually hold the office of Mayor of the City of New York.

As a simplification, past models of CRG have assumed that the system's knowledge was complete and correct. It is understood that these assumptions must ultimately be abandoned. This section suggests that their abandonment will likely require that conversation models be able to indicate which aspects of the system's incomplete and uncertain knowledge are relevant to the response being planned.

---

R: No, I left my watch at home.

The point is that the response would be unacceptable had the question been, "What time is it?" Thus the reasoner cannot discard the literal request.

### The completeness assumption

Only in artificially-constrained reasoning domains is the completeness assumption plausible. But the concession that a response planner's knowledge is likely to be incomplete has important implications for the design of the Conversation Modeler and the models it constructs.

As Examples (40) and (41) demonstrated (see page 70), the incompleteness of a respondent's knowledge may lead to an incorrect diagnosis of the questioner's problem, which can in turn affect the response provided. In that example, the respondent's apparent lack of knowledge that `.login` files are typically linked together (and thus not owned by the questioner) leads her to analyze Q's utterances as manifesting a simple misconception (that "dotted" files are handled differently than regular files when changing their access permissions). This is a case in which R both does not know some proposition $P$ and does not know that she lacks that knowledge. In Example (52) the problem is more obvious:

(52) Q: Does UNIX support any sending or receiving files thru protocol transfers?
R: UNIX supports TCP/IP.

In this example, R's knowledge of "protocol transfers" appears to be incomplete, as it fails to include standard dialup-based file-transfer protocols as KERMIT and ZMODEM. (R's response gives no evidence that she knows of anything besides TCP/IP that could be described as a "protocol transfer".) Consequently she is unable to model "protocol transfers" as an expression having more than a single possible referent.[4]

Incomplete knowledge of this sort, in which an agent lacks knowledge and does not know that she does, is an unavoidable fact of life, one that will affect natural-language dialogue systems as well. As in the case of Examples (40) and (41), such gaps may lead to responses that from the respondent's point of view are helpful, informative, and "cooperative", but which might be judged otherwise by the questioner (or by some other, better-informed observer). This is one of the reasons I have argued that, when modeled as a conversational goal, "being cooperative" is best defined from the respondent's perspective.

Analysis of naturally-occurring data also provides evidence that responding agents often are *aware* of the relevant gaps in their knowledge and that this knowledge influences the planning of their responses. We can see this reflected in Example (53).

(53) Q: Do you know if there is any network link to ucbeuler? I want to send some mail over there.
R: The network link is not up yet. Some time in the near future it will be connected to the Berknet, but I don't know exactly when. It is mostly a software change on ucbvax.

Here R explicitly indicates both that she knows a description of a certain piece of information that would be of interest to Q—the date when ucbeuler will be connected to

[4]As an alternative analysis, R might be applying a strict interpretation to the verb "support." Network communication protocols such as TCP/IP typically require explicit support in the operating system kernel itself, whereas application programs such as KERMIT and ZMODEM require no particular accommodations. Under this analysis, R's response suggests that she failed to take into account differences in meaning and interpretation arising from differences in domain beliefs between the dialogue participants. Such failure might be due to a different sort of incompleteness of R's knowledge.

the Berknet—and that she does not actually have the knowledge so described. This example shows how R's incomplete knowledge and her awareness thereof might affect the information content of her response.[5]

The most common form of incompleteness found in the UNIX transcripts derives from the respondent's inability to build a conversation model sufficient to support the planning of what she would consider a conclusive answer. This generally occurs when the questioner either fails to supply enough detail with his question or uses expressions that the respondent is unable to fully interpret. Consider Example (54):

(54) Q: Have we lost the daemon?
R: WHICH ONE?? The line printer daemon (lpr doesn't work)? The cron daemon (your at files don't get atrun)? The mail daemon (biff doesn't inform you of new mail)?

Here Q employs a definite referring expression which R, given her more detailed knowledge, sees as ambiguous. R is able to recognize that she does not have enough information to properly respond and goes on not only to request further details, but also to provide Q with a space of possible replies and criteria for selecting among them. It is significant that even though R cannot directly address Q's question, her response is cooperative nevertheless. By using a definite referring expression, Q indicates that he believes that there is only one "daemon." In view of that, a response of "Which one?" alone would be uncooperative, since it provides no assistance to Q in answering a question that he clearly does not know how to answer unaided. By knowing *that* more information is needed, and by being able to characterize the *kind* of information needed, R is able to respond in a way that advances the goal of the conversation—resolving Q's UNIX-related problem.

### The correctness assumption

The issue of whether an agent's knowledge is *correct* is also relevant to the specification of the conversation model. Obviously, a respondent who holds incorrect beliefs but believes them to be correct (analogously to a respondent who does not know $P$ and does not know that she does not know $P$) is likely to provide all manner of unhelpful, false, and confusing responses. Once again, there is very little that can be done about this, except to continue to confine such people to working on taxpayer-assistance hotlines.

However, it is clear from available data that, in planning their responses, respondents often attach less than complete certainty to some of their pertinent beliefs and indicate as much when making reply. Consider Example (55).

(55) Q: I cannot find the manual for kermit. What is it? What does it do? Where can I find more information about it? As far as I know, it has something to do with data communication.
R: So far as I know, there is no kermit software on cory. Do you know anything different? P.S. If we don't have the software, it is unlikely we would have the manual.

[5] In a related vein, Mays's work demonstrated that a respondent's belief that a certain event will or might possibly occur in the future could influence her choice of response [Mays 84b].

R's hedges *So far as I know*, *Do you know anything different?* and *it is unlikely* all indicate that R is not certain of the correctness of her belief that there is no KERMIT software on cory. If we suppose that R's belief that "there is no KERMIT software on cory" is the result of a *default inference* [Reiter 80] of the form, "if R is not familiar with a piece of software, then assume it is not installed on the system," then the default nature of its derivation might be the source of R's uncertainty. Example (56), from the EMACS transcripts, provides further evidence for abandoning the correctness assumption.

(56) Q: What is the character to search for if I want to locate a CR using a simple search?

R: Normally you don't. I'm not 100% sure of this, but you may be able to "quote" a CR in a search. You'd do that by being in a search and typing ctrl-Q (for Quote the next character) CR (the actual return key). I don't guarantee this one. What do you want to search for? Normally there are other ways around this.

The response plainly shows that R knew her belief of the ability to quote carriage-return characters while in search mode was less than certain. She felt that the information was useful enough to contribute, but also required a disclaimer.

#### Summary

As we develop natural-language dialogue systems that are capable of increasingly sophisticated techniques of cooperative interaction, we will have to recognize that their knowledge of the domain of discourse is likely to be incomplete, and that some of it may be uncertain as well. To the extent that assumptions of completeness and correctness are avoided, conversation models should be able to indicate both what knowledge pertinent to the response might be absent and which beliefs might be held with less than full certainty. As we saw from the examples discussed in this section, both R's knowledge and her meta-knowledge play important roles in the planning of cooperative responses.

### 4.2.5 Concluding remarks on the Conversation Modeler

This section characterized the general role and theoretical status of the Conversation Modeler in a CRPS. The Conversation Modeler builds the *conversation model*, a special database of beliefs relevant to the conversation providing the evidence that motivates the formation of the system's initial set of cooperative response goals. In defining the Conversation Modeler I am claiming that the process of building this database is separable from response planning.

The Conversation Modeler comprises only those structure-building resources that are routinely applied prior to response planning. Although determining the necessary and sufficient set of such resources is an open research problem, I offered the following pair of conditions as a rough decision procedure:

**(C1)** The procedure must produce information necessary for a CRPS to identify the range of goals it needs to satisfy in order to produce responses that are adequately cooperative.

**(C2)** There is no reasonable way to decide invoke the procedure on the basis of other evidence that could be found in the conversation model.

While conversation models may ultimately need to contain a wide variety of beliefs, I suggested that they must contain at least the following elements in order to support CRG:

- a representation of the questioner's query;
- a representation of the top-level goal the questioner is pursuing that is relevant to the exchange;
- a representation of the plan of action that the system believes the questioner intends to execute; and
- a set of comments representing the system's evaluation of that inferred plan.

In natural-language dialogue systems of realistic size, the conversation model will be a repository for many different sorts of beliefs. The extent to which those beliefs are assumed to be *complete* or *correct* will profoundly affect the system's ability to provide useful and appropriate responses. I have argued that, to the extent that completeness and correctness assumptions are abandoned, conversation models will need to be able to indicate both what knowledge pertinent to the response might be absent and which beliefs might be held with less than full certainty.

## 4.3 The Response Planner

Once the dynamically-constructed portion of the CRPS's environment has been generated by the Conversation Modeler, control is passed to the *Response Planner.* This component of the CRPS has three main responsibilities:

1. Using evidence it finds in the conversation model, it identifies high-level response goals for the system to consider pursuing.
2. It applies context-dependent appropriateness filters to choose a subset of those goals to adopt and plan to satisfy.
3. On the basis of its success at acquiring necessary information and finding methods to satisfy its chosen goals, it constructs a response plan aimed at conveying the information that was found or inferred.[6]

The terminal nodes of the response plan—called the *surface goals* (see Section 4.3.2)—constitute the data from which the Language Generator computes the natural-language response.

The Response Planner is the center of cooperative reasoning and decision making in a CRPS. It is not yet possible to develop a fully fleshed-out specification of this component because, as we will see, such an effort would require a prescriptive theory of cooperative response behavior as its foundation—the very sort of theory that has yet to be formulated (and whose formulation the present work hopes to foster). Instead, arguing from a number of observations about the properties of naturally-occurring cooperative response data, I will describe an architectural framework that seems to capture the gross computational

[6]We will also see that the Response Planner's goals may change over the course of its operation.

characteristics of the process of cooperative response planning. The architecture necessarily leaves much room for refinement, and raises many questions. I will summarize the major open research problems in Section 6.2.

An overview of the Response Planner's architecture is presented in the next section. Subsequent sections will elaborate the theoretical concerns driving its design and sketch its components and their interactions.

### 4.3.1 Response Planner architecture

The architecture of the Response Planner is shown in Figure 4.3. Arrows indicate data flow among its components. The boxes labeled **Conversation Model**, **Response Goals**, and **Response Plan** represent shared knowledge bases that are conceived of as *blackboards*.[7] The shaded elements labeled **Response Goal Proposer** and **Strategic Planner** represent the two active components of the Response Planner. A brief summary of these elements follows.

**Conversation Model** This knowledge base initially contains the collection of beliefs derived from the questioner's utterances by the Conversation Modeler. During the Response Planner's operation, new beliefs may be added to this collection by the Strategic Planner.

**Response Goal Proposer** The **Response Goal Proposer** is the source of the Planner's goals. Drawing on a database of rules, it uses the contents of the Conversation Model blackboard as evidence for response goals the planner should adopt and pursue. The resulting set of goals is written to the **Response Goals** blackboard. Updates to the Conversation Model by the **Strategic Planner** may cause the Response Goal Proposer to modify the contents of the Response Goals blackboard. The operation of the Response Goal Proposer will be discussed in more detail in Section 4.3.4.

**Strategic Planner** The **Strategic Planner** attempts to find methods by which the goals proposed by the Response Goal Proposer could be achieved in a natural-language response. Its output—the *response plan* (Section 4.3.3)—is written to the **Response Plan** blackboard. Through a process called *reflection* (Section 4.3.6), results of the Strategic Planner's operation may be fed back into the Conversation Model database, enabling the Response Goal Proposer to decide to modify the system's set of adopted goals. Issues pertaining to the design of the Strategic Planner will be discussed in Section 4.3.5.

### 4.3.2 Surface goals

Before explaining the Response Planner's architecture, I would like to further examine the goal-based character of natural-language responses. This section presents a theory in which responses are modeled as sets of *surface goals*: goals that responses attempt to achieve by way of their particular linguistic form and content. I will argue in the next section that if we are to develop accounts that explain *why* particular surface goals are adopted in a given query situation, we must attempt to connect the surface goals to more general *response*

[7] A survey and discussion of blackboard architectures and blackboard-based systems can be found in Nii [Nii 86b, Nii 86a] and Cheikes [Cheikes 89].

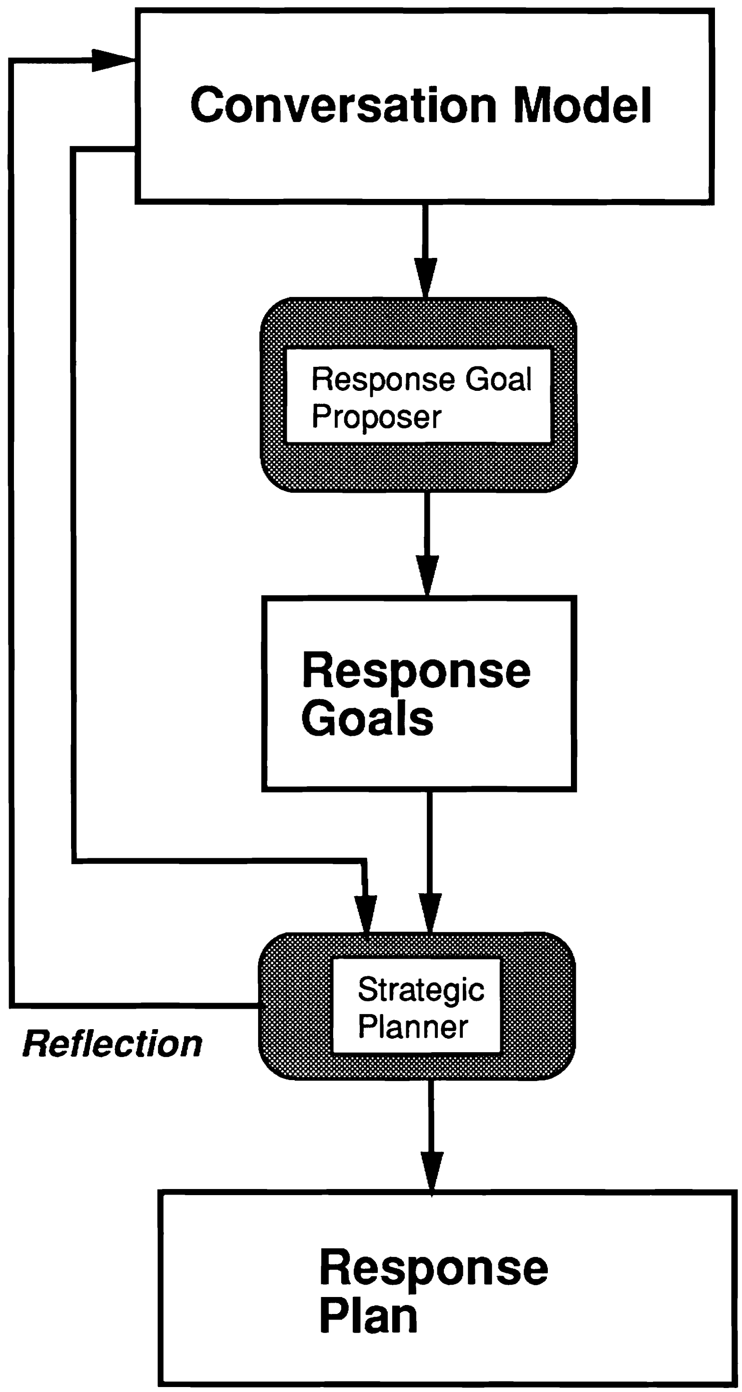


Figure 4.3: Architecture of the Response Planner

*goals* whose adoption is motivated by principles of cooperation. This theoretical model of responses is fundamental to the Response Planner's design.

Data obtained from studies of naturally-occurring question-response pairs has provided a foundation for the development of numerous models of cooperative interaction. In the analyses performed over the course of the research being reported here, the notion of a *surface goal* has served as a useful abstraction. I will develop a characterization of surface goals with the aid of Example (53)—repeated below as (57)—taken from the UNIX transcripts.

(57) Q: Do you know if there is any network link to ucbeuler? I want to send some mail over there.

R: The network link is not up yet. Some time in the near future it will be connected to the Berknet, but I don't know exactly when. It is mostly a software change on ucbvax.

The view that response generation is a planning process means that a respondent forms a reply with certain desired effects (alterations to the questioner's belief space) in mind, and that bringing about those effects is the driving force behind her choice of the particular linguistic content and form of the response's constituent utterances. Given that, it is reasonable to conceive of the response that is actually provided as indicating what the respondent believed to have been her most reliable means (given the constraints on the planning process and considerations of diminishing marginal returns) of achieving her ends in the particular context of utterance.

As a research method, it makes sense to examine naturally-occurring response data and hypothesize the effects that different responses might have been intended to achieve. The most reliable sorts of hypotheses are those that are based on the literal meaning of the constituent utterances of a response. For example, we could decompose the response in (57) into the following four utterances:[8]

1. "The network link is not up yet."
2. "[Ucbeuler] will be connected to the Berknet sometime in the near future."
3. "I don't know exactly when [ucbeuler will be connected to the Berknet]."
4. "[Connecting ucbeuler to the Berknet involves] mostly a software change on ucbvax."

We might then model the literal meaning of these utterances in this way (where the representation is meant only to be suggestive):

1. `¬connected(ucbeuler,berknet)`
2. `occur(connect(ucbeuler,berknet),near_future)`
3. `¬know(R,[the ?t:time . occur(connect(ucbeuler,berknet),?t)])`
4. `enables(upd_software(ucbvax),connect(ucbeuler,berknet))`

[8] Brackets are used either to indicate replacements of pronouns by their intended referents or to surround text realizing concepts that appear to have been elided in the original response.

If we assume that R's assertion of a proposition $P$ constitutes sufficient evidence for ascribing to R an intention that Q believe $P$—an entirely reasonable assumption in the context of the UNIX consultant dialogues—then for each $P$ above, we could ascribe to R an intention that `BEL(Q,P)`. I will label these intentions the *surface goals*, for they are the goals that the constituent utterances of a response will represent attempts to achieve. It should be borne in mind that surface goals represent only *attempts* to affect the questioner's mental state. A respondent can have no assurance that her utterances will actually achieve her surface goals. (Misunderstandings arising from failures of this sort may cause questioners to initiate clarification subdialogues.)

Other goals besides surface goals will typically participate in the choice of a response's particular linguistic content and form. At the stylistic level, unusual conciseness of expression might be used to convey an air of expertise, superiority, or impatience.

(58) Q: Where have the online messages gone?
R: They've been fixed.

Recondite lexical or grammatical choices might be employed to suggest erudition. In this vein, Hovy has investigated different kinds of interpersonal goals that influence natural-language generation [Hovy 88], and Rubinoff is exploring the thesis that the language generation process should be viewed as an attempt to satisfy goals at multiple levels [Rubinoff 90]. But computational accounts of how language can be used to achieve goals, by subtle means or blatant, have as yet barely scratched the surface.

I am therefore restricting surface goals to include only those desired alterations of the hearer's mental state that can be effected straightforwardly by the linguistic form and content of the utterances of the response. I consider the development of principled accounts linking properties of query situations to decisions to attempt to achieve particular surface goals to be the greatest concern of the respondent-based program of research on cooperative response generation.[9]

### 4.3.3 Response goals and response plans

The Response Planner's architecture design represents an attempt to characterize the components and organization of the reasoning process that connects features of the conversation model to selected surface goals that are the basis for the system's natural-language responses. I will now further analyze the structure of that reasoning process, arguing for a distinction among a respondent's general *response goals*, her *response plan* for achieving those goals, and the surface goals she finally adopts and attempts to achieve by utterance acts. This three-way division is embodied in the Response Planner's design.

#### Acting and thinking about acting

People demonstrate the capacity to both act and think about acting. In traditional models of planning, agents search for actions or action sequences they can execute in order to achieve certain goals. Their goals originate from outside the system, for example, from commands given by a user. "Thinking about acting" in this context means reasoning

[9]In the remainder of this thesis, I will exclude from consideration all physical elements of responses, such as physical actions, gestures, facial expressions, and intonation.

only about how actions can be used to bring about desired effects in a given execution environment.

Cooperative response planning systems, unlike more conventional planners, reason not only about how to satisfy their goals but also about *what goals they ought to satisfy.* That is, they "think about acting" at more general levels than conventional planners are required to. For example, given a questioner's request for information, cooperative respondents are able to reason about whether to satisfy the request. This is a decision process qualitatively different from either identifying the information that would have to be provided in order to satisfy the request or computing a set of utterance acts that would serve to convey that information to the questioner.

To elaborate on this point, let us return to Example (57) and consider a possible account for R's decision to adopt the first surface goal:

`BEL(Q,¬(connected(ucbeuler,berknet)))`

Suppose we were to treat the act of satisfying this goal as corresponding roughly to the act of giving the *direct answer* to Q's query—the information explicitly requested ("Do you know if there is any network link to ucbeuler," ignoring its interpretation as a request for information about R's knowledge state).[10] We might envisage the process by which R decides to adopt this goal as proceeding through the following steps:

**Step 1**: R determines that Q wants to know if there is a network link (from the current machine) to ucbeuler;

**Step 2**: R adopts a goal $g$ that Q know whether there is a network link to ucbeuler;

**Step 3**: To satisfy that goal, R consults her own knowledge base and finds that she believes that there is no network link to ucbeuler;

**Step 4**: R decides that she can satisfy goal $g$ by satisfying goal $g' =$ Q believe that there is no network link to ucbeuler;

**Step 5**: R decides that she can satisfy goal $g'$ by satisfying goal $g'' =$ Q believe that ucbeuler is not connected to the Berknet.[11]

Step 2 is the step of primary interest here. I will use the term *response goal* to denote such general goals as "Q know whether proposition $P$ is true." They are abstract goals, motivated by evidence in her conversation model (in this case, her interpretation of the intentions underlying Q's question), that R adopts and plans to achieve. As a result of that planning activity (partially illustrated above), R selects surface goals which are then used to guide the construction of a natural-language response.

My claim is that such general goals must be *explicitly represented and reasoned about* during the planning of cooperative responses. The primary theoretical inadequacy of questioner-based models is their failure to model these goals and connect them in principled ways to the utterances that are used to achieve them.

In contrast, one of the central concerns for a respondent-based account of CRG is modeling the decision process that takes place between the first and second steps above.

[10] R's use of "yet" in the actual data suggests that a proper analysis is more complicated than this, but a more detailed analysis would not invalidate the point.

[11] Uttering "The network link is not up yet" is viewed (roughly) as a linguistic act chosen as a method for achieving the (surface) goal decided upon in Step 5.

That is, we must answer the question: *why* does R choose to adopt response goal $g$, and in particular, how does the decision to adopt that goal follow from principles of cooperation?

Clearly, the mere fact *that* Q asked the question does not in general constitute sufficient justification for R's decision to adopt a goal of answering it. Rather, it can only provide a basis for a decision by R to *consider* adopting that goal. This idea is supported by Pollack's research demonstrating that respondents' beliefs about the well-formedness and executability of questioners' domain plans serve as an appropriateness filter on direct answers [Pollack 86].

Other appropriateness filters exist as well. Suppose, for example, that R believed Q's plan to be ill formed, for example, she believed that it was unnecessary for ucbeuler to be connected to the Berknet in order for Q to send mail there.[12] In such a situation, R might very well be inclined to disregard the question. In a similar vein, the response below provided by a different consultant suggests that R was able to decide that the issue of whether euler is connected to the Berknet was largely irrelevant to the task of helping Q send mail to users there.

(59) R: Just do this: mail ruby!euler!(login name). Let us know if it doesn't work. Euler is only reached thru the ruby machine.

The above response sidesteps the question of euler's network connectivity entirely, focusing instead on how Q might send mail to users at euler. All this argues strongly for the claim that cooperative respondents must be able to distinguish response goals *under consideration* from those they actually adopt and attempt to satisfy.

Differences like those between the two responses in (57) and (59) pose a challenge for models of cooperative response generation. If we assume that both respondents were behaving "cooperatively" to the best of their respective abilities, then the difference in response ideally should be traced to differences in their respective beliefs rather than differences in their competence as cooperative dialogue participants. For example, in (59) we would like to claim that R responded as she did because she had knowledge of a way to help Q achieve his goal, which enabled her to focus on that in her reply. We do not, however, want to ascribe that same knowledge to R in (57), for in that case her response would appear to be uncooperative, as it would then be ignoring Q's desire to send mail. The challenge is to devise a single model of CRG, sensitive to these differences in belief, that accounts for both responses.

Distinguishing a level of reasoning about general response goals promises to help us meet this challenge. For example, to reach a unified model of the responses in (57) and (59) we might hypothesize that both respondents, in virtue of their respective models of the conversation, choose to consider the same two response goals: the first, to provide an answer to Q's query, and the second, to enable Q to send mail to euler. The first respondent finds that she cannot satisfy the second goal, and thus plans a response that provides an elaborate satisfaction of the first goal. The second respondent, however, recognizes that not only can she satisfy the second response goal, but in so doing she need not satisfy the first (or, alternatively, satisfying the second will indirectly satisfy the first).

Once a respondent adopts a response goal, she must proceed through a planning process to compute *how* to achieve it in a natural-language response. In the framework I am

[12] Pollack would call this an *incoherent* plan [Pollack 86].

developing here, this process results in the selection of a set of surface goals which the Language Generator can use to pick appropriate natural-language expressions. I will use the term *response plan* to refer to the structure linking response goals to the surface goals that together should bring about their satisfaction. Building this structure is the Response Planner's function. In the next section, I will look more closely at the process of proposing response goals. Section 4.3.5 will then consider how a CRPS might construct response plans.

#### Summary

I have developed a three-tiered theory of response structure that distinguishes the respondent's high-level response goals, her response plan for achieving those goals, and the surface goals that are the terminal nodes (the executable actions, in some sense) of that plan. Analysis of naturally-occurring response data suggests that a theory of cooperative response planning ought to account for the apparent capacity of respondents to reason about and select among different potential response goals. For example, respondents must be able to decide not only *how* to answer a question, but also *whether* to answer it. Explicitly embodying such a distinction in a framework for cooperative response planning should help researchers to develop accounts of alternative responses to queries while maintaining consistent sets of assumptions about the respondents' basic cooperative competence.

### 4.3.4 Choosing response goals

In the architecture shown in Figure 4.3 (page 92), the *Response Goal Proposer* (RGP) is the component responsible for providing response goals for the system to pursue. Its primary source of input is the *Conversation Model*, a shared knowledge base whose initial contents consist of the data generated by the Conversation Modeler. (The "reflection" feedback loop, to be discussed in Section 4.3.6, may lead to updates of the Conversation Model during the response planning process.) Its output, a set of response goals, is written to the *Response Goals* panel, another shared knowledge base.

#### Principles of operation

Response goals are proposed by way of a two-step procedure, as follows:

1. Features of the conversation, as recorded in the Conversation Model, provide evidence for the *consideration* of one or more response goals.

2. For each response goal under consideration, context-dependent *screens of admissibility* (to be discussed in this section) are used to decide whether the goal should be adopted.

Response goals that satisfy all applicable screens of admissibility are written to the *Response Goals* blackboard. At that point, they become the system's adopted response goals. Although they will all be motivated by features found in the conversation model and will have passed all applicable screens of admissibility, they will not necessarily have been tested for *achievability*. (I will elaborate on this point shortly.)

In the previous section, I argued that a computational model of cooperative response planning should incorporate a notion of high-level response goals distinct from surface goals; I did not, however, actually define what response goals are. In fact, this is very difficult to do without a better-developed theory of cooperative behavior than we have at present. The intuition is that response goals are high-level goals to bring about certain desired changes to a questioner's beliefs. Specific conditions of a conversation provide evidence for the adoption of various general goals; once adopted, plans tailored to the context must be devised to achieve them.

To clarify this point, we once again consider the case of misconception correction. We might imagine in the context of a particular CRPS application assigning response-goal status to the general activity of "correcting an object-related misconception" [McCoy 88]. McCoy never addressed this issue, but there is presumably some well-defined set of detectable features of a conversation that indicate whether the questioner has an object-related misconception. Suppose tests on the model of a particular conversation were to reveal that Q holds a misconception of the form `(ISA X Y)` for some discourse entity $X$ and domain concept $Y$. This could serve as evidence for R to consider a response goal of correcting that misconception, which might be modeled as the goal `(BEL Q (NOT (ISA X Y)))`. In this simple view, misconception-correction behavior is seen as following from a goal of causing the questioner to believe $\neg P$ for some proposition $P$ such that both R believes that Q believes $P$ and R believes $\neg P$.

In order to accomplish this general goal in a particular discourse setting, R might have to provide an extended response along the lines of those that McCoy's ROMPER constructs. The response acts of denying that $X$ is a kind of $Y$, conceding their similiarities, and asserting their differences might all play important roles in achieving the high-level goal of belief alteration. But the decision as to whether to use those acts or others will in general be sensitive to such factors as variations in discourse context and R's beliefs about the questioner's domain goals. Thus ROMPER illustrates one method for achieving a general goal of misconception correction under a particular set of conversational circumstances.

I envisage the RGP as having access to a library of response goal "templates", each capturing the planner's knowledge about a general class of goal-driven cooperative response behavior. Besides goals having to do with misconception correction, other examples of possible response goal classes could include:

- give the direct answer to a yes/no or wh-question;
- explain how to perform a particular step in a plan;
- cause the questioner to believe his plan is unexecutable;
- explain why the questioner's plan is unexecutable;
- cause the questioner to believe his goal cannot be achieved at the present time;
- suggest an alternative course of action;
- describe a new plan of action.

Determining an appropriate level of detail for the definition of response goal templates in general is a research question; the examples above are meant only to be suggestive.

In fact, defining the space of response goal classes—if such a space is indeed finite—will prove to be a critical step in the effort to formulate a theory of cooperative behavior within the architectural framework I am proposing. However, for the purposes of using this architecture as a foundation for the implementation of a practical natural-language dialogue system, it should be possible to derive a reasonable set of response goal classes from the domain of discourse.

Each response goal template must have associated with it a set of tests on the conversation model to be used to determine when an instance of the response goal should be considered for adoption. I will henceforth call these the response goal's *activation conditions.* Response goal templates are essentially descriptions of high-level actions that the planner knows how to plan to achieve. Activation conditions, however, are not exactly analogous to action preconditions; preconditions characterize states of the world in which a particular action is actually executable, whereas activation conditions only characterize states of a conversation model (representing the respondent's understanding of the questioner's utterances) in which a particular high-level goal is considered to be worthy of adoption.

During the first step of the RGP's operation, it scans through its library of response goal templates and identifies all the goals whose activation conditions are satisfied. These goals are then instantiated (receiving appropriate bindings from the conversation model) and become the RGP's *potential response goals.*

The potential response goals are not written to the Response Goals blackboard, but rather are held internally by the RGP for use during its second step of operation. As I argued in Section 4.3.3, the fact that a response goal's activation conditions are satisfied is not in itself sufficient grounds for a respondent to decide to actually adopt that goal, e.g., the detection of a misconception need not necessarily lead to a correction in the response. Rather, there appear to be sets of what I will henceforth refer to as *screens of admissibility* that are used to rule out the adoption of activated response goals whose achievement in a given discourse context would lead to conflicts with either the respondent's higher-level discourse goals or the general principles of cooperative interaction that underlie her behavior.

For an example of a conflict with the respondent's higher-level discourse goals, we recall Example (35)—repeated below as (60)—in which we saw that a respondent might decide not to adopt a goal of misconception correction because it interfered with the achievement of other more important goals.

(60) Q: Do I want a mark at the beginning and at the end of the block?
R: For right now, just go ahead and set a mark any old place. (Just to see how it works.)

In explaining this example, I hypothesized that R was focusing on her pedagogic goal of teaching Q how to set the mark in EMACS; in virtue of that she might have decided that a misconception correction at this stage of the dialogue would not be helpful.

With respect to conflicts with principles of cooperation, I have observed that Pollack's research implies that respondents seem to have the ability to make decisions to omit direct answers based on their beliefs about the well-formedness and executability of the questioner's domain plan [Pollack 86]. In this case the respondent's decision might be driven by some underlying principle of parsimony; in the circumstances studied by Pollack, the answers to the queries are rendered relatively useless by the ill formedness or unexecutability of the corresponding explanatory plans. That is, satisfying the goal of answering the question might be seen as conflicting with the principle that the respondent not provide information that does not in some way advance one of the questioner's goals.

Finally, some screens of admissibility may involve the consideration of possible interactions between potential response goals. For example, in trying to account for R's response in Example (59), it may be necessary to propose that R be able to perform the following inference chain: Q asked his question (requesting information on the state of network connectivity to euler) in order to find out how to send mail to euler. If R is able to satisfy the (potential) response goal of enabling Q to send mail to euler, then it would not be necessary to also satisfy the goal of answering the query. That is, R may have to evaluate how the successful accomplishment of one response goal affects the need to attempt others.

Clearly, there is much work remaining to be done to develop a good characterization of screens of admissibility. If cooperative respondents can truly be modeled as selecting response goals based on their general understanding of the questioner's utterances, then examples of naturally-occurring responses seem to indicate that respondents do not always act on all the goals that appear to be well motivated given the questioner's utterances. Understanding cooperative response behavior demands that we understand both why some goals are chosen and why others are disregarded.

Fleshing out the notion of screens of admissibility, understanding how they are related to activation conditions, and understanding how both are connected to (or derived from) principles of cooperation must be left for future work. For the present, it is useful to recognize these problems as arising from the attempt to reformulate cooperative response behavior in a goal-based framework.

**Testing the achievability of response goals**

I mentioned earlier that the RGP does not necessarily verify that the response goals it proposes are in fact achievable. That is, the theory underlying the architecture permits the RGP to suggest goals that the system might later find to be unachievable. I will motivate that idea now, arguing that it is supported by examples such as (61)–(63).

(61) Q: Could someone please tell me where I might get on-line documentation on the EMACS editor?
R: I know of no online documentation.

(62) Q: I tried to send mail to someone in "lll-mfe" and mail said that the service was not available. Why not?
R: What was the exact mailing address you used? You may have used the wrong syntax for the network you wished to use, or you may need to specify intermediate site names.

(63) Q: For some reason, my /tmp write permission seems to have been lost in the wind. Help! I can't do anything without! Would you know what's wrong?
R: I'm sorry but I can see no reason why. Can you be more specific? The only case like this I've seen is someone who set his umask to 700. Did you do this?

I have claimed that a respondent-based view of cooperative response planning demands that the respondent be able to form high-level response goals and evaluate the appropriateness of adopting them in a given query situation. In (61), for example, R could be supposed to form and consider adopting a response goal that is something like "enable Q to have access to on-line EMACS documentation". The motivation for this goal would come from R's model of the conversation indicating Q's desire for EMACS documentation.[13] In this case, however, R discovers that, although it would certainly be appropriate on grounds of cooperativeness to satisfy that goal, she finds that she cannot. It is her awareness of her inability to satisfy the proposed response goal that leads her to adopt new goals (by means of *reflection*—Section 4.3.6), in this case, to indicate (indirectly) that she cannot help.

In Example (62), we might hypothesize that R was attempting to satisfy a response goal of the form, "Q understand why mail to lll-mfe failed". In trying to satisfy that goal, R discovered she did not have enough data to construct an adequate explanation. Her recognition of this situation then led her to adopt and pursue goals having to do with acquiring the needed information from Q (and explaining her reasons for asking). Example (63) is similar, where R's apology may be taken as evidence that she recognized her inability to satisfy a response goal adopted on Q's behalf.

The important conclusion to be drawn is that naturally-occurring data suggests that respondents carry out diverse reasoning activities in the pursuit of their response goals. They focus on particular tasks they need to accomplish, and their knowledge of their success or failure in performing those tasks influences their subsequent reasoning and the choices they make in forming their responses. The Response Planner architecture design attempts to model these activities.

The principle that RGP-proposed goals are not guaranteed to be achievable suggests the following view of goals on the Response Goals panel: they are general goals whose satisfaction in the current query situation is warranted on the basis of principles of cooperative interaction. A respondent's inability to fully or partially satisfy any or all of them may provoke her to adopt new goals.

### Summary

This section outlined the general theory of the Response Goal Proposer, the source of goals in a Cooperative Response Planning System. The process of proposing goals proceeds in two steps. First, drawing on a library of response goal templates, the RGP identifies potential response goals whose activation conditions are satisfied. The resulting set comprises all response goals for which features of the conversation provide minimal justification for adoption. More general sets of appropriateness tests, called *screens of admissibility*, are then used to decide which potential response goals actually ought to be adopted and which can or should be disregarded. Determining the relationship between activation conditions, screens of admissibility, and principles of cooperation is a challenging task for future research. Potential response goals that pass all applicable appropriateness tests are written to the Response Goals blackboard and are thereby adopted. The RGP does not test these goals for achievability, and it may later be discovered that some or all of them cannot be

[13] Note that, if Q had provided additional information that suggested an answer to the query was not useful, e.g., "I typed 'emacs' and got the message 'Permission denied'", then the answer-query response goal could be eliminated from consideration.

reached. A feedback process called *reflection* (Section 4.3.6) exists to permit the RGP to modify the system's set of adopted goals during the response-planning process.

### 4.3.5 Planning to achieve response goals

When the RGP completes its work, the Response Goals blackboard of the Response Planner will contain a set of high-level response goals that have been judged to be worth pursuing based on information found in the conversation model. The task of the *Strategic Planner* (SP) is to compute a set of surface goals whose achievement through the system's natural-language response would have the effect of satisfying the system's high-level response goals.

In trying to design a working prototype of the SP, researchers will be forced to confront a plethora of difficult research problems all related to this question: How do agents use language to achieve their goals? While it has long been a widely accepted principle that language use is a form of action, only recently have researchers begun to investigate the planned nature of natural-language generation [Appelt 85b, Hovy 88, Moore 89, Rubinoff 90]. Because our comprehension of both the planned nature of language and the nature of high-level response goals is currently quite limited, the discussion in this section must be confined to a few general observations about the issues that a design for the Strategic Planner ought to address.

Based on the analyses from which the overall architecture was derived, the two most important considerations in the SP's design are these:

1. The SP is a *conjunctive goal planner*, that is, it computes plans that are intended to satisfy multiple goals.

2. The SP may be unable to find a plan that satisfies some or all of the system's adopted goals. The overall Response Planner needs to be able to adjust its behavior in some appropriate way when that occurs.

I examine these issues next.

#### Planning for multiple goals

Through the analyses of naturally-occurring response data that have been discussed so far, I have tried to argue that rich responses of the sorts that we find in the UNIX and EMACS transcripts are best seen as derived from plans to satisfy several high-level response goals. In one analysis of Example (57), for example, I suggested that the conversation model could provide evidence for the adoption of at least two high-level goals: (1) that Q *know whether* a network link exists between the current machine and ucbeuler, and (2) that Q *know a plan* for sending mail to users on ucbeuler. It would be illuminating at this point to describe the analytic process that led me to this hypothesized pair of goals.

We begin by sketching a model—the conversation model—of the beliefs that R could reasonably be expected to ascribe to Q on the basis of the observed utterances. For the present example, R could be expected to attribute at least the following beliefs to Q:

1. Q's top-level goal is to send mail to (unspecified) users of ucbeuler;

2. Q wants to know if a network link exists between his machine and ucbeuler;

3. Q believes that the existence of a network link between his machine and ucbeuler is an enabling condition on the action of sending mail to users at ucbeuler;

4. Q's goal in asking the question is to determine whether there exists a valid plan for sending mail to users at ucbeuler.

R gives no indication in her response that she believes any of the beliefs ascribed to Q to be incorrect. Because she focuses on the state of the network link, we may deduce that she held the following beliefs prior to the conversation:

1. In order to send electronic mail between machines A and B, a network link must exist between A and B.

2. No network link exists between Q's machine and ucbeuler.

3. The state of network connectivity between two machines may change over time.

4. Plans are afoot to establish a network link between Q's machine and ucbeuler.

The question that remains to be answered is this: how might a cooperative agent reason from all these beliefs to arrive at a set of response goals?

Since we lack a theory of cooperation from which to rigorously derive a choice of high-level goals, the best we can do at this point is make some reasonable assumptions. For whatever goals we pick, the principles underlying their selection should reflect some general characterization of cooperation, even if rough and incomplete. It is crucial that we avoid the trap of allowing our theories to be strongly influenced by the details of particular responses: that would only lead us to pick goals that reflect the state of R's reasoning after most of the important cooperation-based decisions have been made. The point of this exercise is to develop a better understanding of R's reasoning process, not to model particular responses as sets of goals. Thus we want to choose goals that are grounded in some independent theory of cooperative behavior.

An oft-repeated characterization of cooperative agents is that they act to enable questioners to achieve the questioners' goals. UCEGO, for example, upon detecting that the user wants to perform an action $A$, adopts the goal of enabling the user to do $A$ [Chin 88a]. In [Cohen 90b, p. 229], Cohen and Levesque characterize cooperative agents as "sincere and helpful... adopting someone else's beliefs and goals else as [their] own". The claim behind Allen's thesis, that "many instances of helpful behavior arise because the observing agent recognizes an obstacle in the other agent's plan, and acts to remove the obstacle" [Allen 83, p. 108], is an extension of this basic idea. As a theory of cooperation, goal-enabling agency is a simplistic one and suffers from the same deficiencies of questioner-based theories that were discussed in Chapter 3. The fact that cooperative agents often seem to act in this way is surely epiphenomenal. Nevertheless, such a theory does have broad explanatory power and is useful at least as a jumping-off point to more satisfactory accounts.

If cooperative agents act to help questioners to achieve their goals, then we may assume that should R determine that answering Q's query might help him to reach his goal, she would adopt the high-level goal of providing that answer. This leads to the first response goal, that Q know whether there is a network link between Q's machine and ucbeuler.

Cooperative respondents also seem to be able to recognize when an answer by itself would be inadequate as a response. If R had reason to believe that Q knew how to send mail between network-connected machines and thus was only querying ucbeuler's connection status, she might be able to decide that just a direct answer is required (although this decision is affected by whether the answer is *yes* or *no*). But in the absence of such a belief, more general goal-enabling behavior may be required. I accommodate this idea by positing R's adoption of a second explicit goal of enabling Q to achieve his stated goal.

Analyses of this kind that I have performed on other examples from the transcripts (assuming a Conversation Modeler as capable as the one described in Section 4.2.3) have generally supported the conclusion that in most query situations, several high-level response goals can usually be derived from general cooperative principles (even simple ones). This tells us that the Strategic Planner should be able to devise plans to achieve multiple goals.

Conjunctive goal planning has always been of interest to the planning research community (see [Chapman 87] for one formalization of the general problem). This is a considerably more complex task than single-goal planning, and engenders many difficult problems. Two important questions for future research regarding the SP's design are: Could the RGP ever propose goals that "conflict" in some sense, and if so, what sorts of conflicts might arise, and how could the SP detect and deal with them?

### Unachievable or abandoned goals

I argued in Section 4.3.4 that it is reasonable to expect the RGP on occasion to propose goals that later turn out to be unachievable. That idea is both reinforced and refined by a comparison of the reasoning processes that seem to be needed to produce the responses in Examples (57) and (59).

R focuses in (57) on ucbeuler's network connectivity; in contrast, her focus in (59) is on providing information that enables Q to achieve his top-level goal. My contention is that, under certain assumptions, both responses can be considered cooperative. In (57) we must assume that R does not know of any way for Q to get mail to euler, because if she did, her response would violate our intuitive conception of cooperative behavior. In (59), we must assume that R determines it is unnecessary to respond to the question of euler's network connectivity in view of her ability to satisfactorily address Q's top-level goal.

Our identification of these assumptions exposes important dimensions of cooperative agents' reasoning abilities. In devising an account for the response in (57), we would like to understand the role that R's awareness of Q's top-level goal plays in her decision-making process. Similarly, we would like to understand the role that R's awareness of Q's explicit request in (59) plays in her decision making. Of course, I am assuming here that R does in fact explicitly reason about these matters, but any other view seems unmotivated.

With these considerations in mind, I have proposed an analysis in which R's underlying principles of cooperation lead her to adopt the same pair of high-level goals in each situation, one that would give rise to some form of direct answer and the other that would give rise to some form of goal-enabling behavior. This analysis then forces us to examine how R might proceed from these two goals to reach either of the two actual responses.

While it is useful to tease apart the steps of R's reasoning process in this way, my analysis nevertheless raises computational difficulties. Consider that a satisfactory account of (57) needs to answer at least these questions:

- How precisely does the SP determine that it cannot help Q to achieve his goal?

- How could this determination affect decisions that are made in the planning for the other goal? For example, might R have decided to provide an elaborate response focusing on euler's soon-to-be-established network connection in order to compensate, in some sense, for her inability to help Q reach his goal?

- For what reason(s) could R have decided that it would be of interest to Q to know that a network connection would be established "in the near future"?

An account of (59) must answer a different set of questions, the primary one having to do with how R could decide that an explicit answer would be unnecessary under the circumstances.

The conclusion to be drawn from the preceding discussion is that the goals placed on the Response Goals blackboard by the RGP should only be regarded as the system's initial selection. As the Strategic Planner operates, it may find that some goals must be abandoned and others modified. This idea will be expanded upon in the discussion of *reflection* in Section 4.3.6.

**Summary**

This section has discussed the two central issues in the design of the Strategic Planner: (1) it must be able to compute plans that satisfy more than one high-level goal, and (2) it must be able to cope with the fact that not all the goals it is given will necessarily turn out to be achievable. This second concern suggests that a channel of communication from the SP to the RGP is required. I take up that matter next.

### 4.3.6 On the need for feedback during response planning

It is between the Response Goal Proposer and the Strategic Planner that we may draw the line separating issues of cooperation from those of language generation. In the framework I have laid out, principles of cooperation guide the system's choice of its high-level response goals. Once the goals are selected, the problem becomes one of constructing natural-language expressions that could be uttered as a means of achieving those goals. However, analysis of naturally-occurring data suggests that the processes of goal selection and plan formulation are not as independent as this division implies.

I argue in this section that the goals proposed by the RGP on the basis of data found in the conversation model are not necessarily the only high-level goals that a CRPS will attempt to achieve through the system's response. Rather, knowledge acquired during the Response Planner's operation may lead to the adoption of new goals or to a revision of the system's set of adopted goals.

These insights have led me to propose a feedback channel connecting the Strategic Planner to the Conversation Model blackboard. This channel enables the SP to dynamically inform the RGP of the results of its operations. This in turn makes it possible for the RGP to update the contents of the Response Goals blackboard as needed in light of new information that becomes available to it. Of course, feedback of this sort poses difficult computational problems that must ultimately be faced. (I will discuss some of these problems briefly at the end of this section.) For the moment, I can discuss only the observations that imply feedback is necessary.

### Dynamic adoption of new goals

One observation implying feedback from the SP to the RGP is that some elements of naturally-occurring responses appear to be triggered not by evidence in the initial conversation model but rather on the basis of reasoning about other response elements. The planner seems to have the ability to keep track of its history of response decisions in computing the current response and to make new decisions "on the fly", driven by reasoning about the consequences of its earlier decisions.

For example, decisions to provide clarifications or justifications may depend on reasoning about other things the planner has already decided to say. This is illustrated in Example (64).[14]

(64) Q: Is there a way to send mail to ucbeuler from ucbcory?
R: Yes, it is letter y on the Berknet. So mail user@y.CC. If you have further problems with it, mail to serge@cory. He is the euler system manager.

After processing Q's utterances and building a conversation model, R could reasonably be expected to adopt such high-level response goals as "Q know whether a plan exists for sending mail from cory to euler" and "Q know a plan for sending mail from cory to euler". But how are we to explain the appearance of utterances "If you have further problems with it, mail to serge@cory" and "He is the euler system manager"?

R's use of the word *further* makes analysis of her utterance of "If you have further problems" difficult. It might be that she inferred or knew *a priori* that Q had been having trouble sending mail. For example, she might have interpreted the fact of Q's request as evidence that he had tried and failed to send mail to euler. On the other hand, R might have been uncertain regarding whether "user@y.CC" was really the correct address; in this case, her reference to "further problems" might mean something like "further problems getting mail to euler, including problems related to the possible inaccuracy of my response".

In contrast, it is quite clear that R's utterance of "He is the euler system manager" is an attempt to justify her suggestion that Q send mail to serge@cory in case of difficulty. R could not have decided to provide that justification without having first decided to mention serge@cory. In other words, R first made the high-level decision of telling Q what to do in case of trouble. When that decision was played out—that is, a plan was formed to carry it out—R saw that it was necessary to mention serge@cory. In view of that, she then decided that it was important to tell Q who serge@cory is so that Q would know why it is reasonable to send requests for help to serge.

Note that R's decision to provide justification is a considered one. An act of justification would not have been necessary, for example, if R believed that Q already knew who serge@cory was (if provided in that case it would be better viewed as a kind of reminder rather than as a justification). Decisions of this sort, however, are the responsibility of the RGP; the SP is only the effector, finding linguistic methods of achieving goals provided by the RGP.

In view of that, the RGP needs to have access to the SP's output. If the RGP were able to inspect the response plan being built by the SP, it could notice that a decision had been made to refer to serge@cory. Then, if so motivated by its rules of cooperation, it could decide to adopt a new goal of providing justification.

[14] This is also Example (37) on page 60.

**Dynamic goal revision**

I have argued that characteristics of naturally-occurring responses such as apologies, attempts to redirect questioners to other information sources, and requests for elaboration or clarification indicate that cooperative respondents, in the course of planning their responses, sometimes adopt goals that during the planning stage are discovered to be either unachievable or only partially achievable. Human cooperative respondents obviously do not "fail" under these conditions; rather, they notice the problem and adapt their behavior accordingly. In order for cooperative response planning systems to have this same ability to notice and adapt to planning failures, a feedback channel is needed between the goal planning and goal selection components.

Let us explore this idea further with the aid of Examples (65) and (66).

(65) Q: Can I find out coral's load average from ucbcory?
R: You would have to login to find it out. They do not broadcast the info to our net.

(66) Q: Is there a way to find out who's on other systems using a modified "f" command? Or, how would you do it?
R: "f," "finger" on onyx are only local. Other machines, like bugs, can finger people on some other machines.

In (65), Q wants to obtain a plan (i.e., be informed of a UNIX command sequence) that he can execute on ucbcory to obtain information about the system load of another machine, in this case, coral. When R attempts to compute such a plan, she finds that one does not exist. But R does not simply say, "No, that is not possible." Instead, she seems to alter her reasoning strategy, presenting Q with a less desirable (indeed, the only) plan and explaining why cory does not have access to coral's load status.

We see similar behavior in Example (66). R presumably begins her cooperative reasoning activity looking for a command that, when executed on the local machine (onyx), provides local users with lists of active users on remote machines that the users specify. When she fails to find such a command, she alters her response goals to focus instead on the fact that the local machine (onyx), unlike other machines, lacks the facilities to perform remote finger operations.

Precise details of how a respondent might actually compute the responses above remain to be worked out; it is clear, however, that as we decompose the process of cooperative response planning into more finely grained, deliberately chosen reasoning steps we will discover that high-level goals assumed to be achievable may later be found not to be. This strongly suggests that cooperative response planning crucially demands an ability to react to planning failures. That is, in the course of cooperative response planning, a respondent might adopt what appears to be a worthwhile goal, then later discover that she cannot achieve it. But such discoveries cannot lead to the breakdown of the overall planning process; rather, the respondent must be able to react to and reason about these internal planning failures and alter her goals appropriately. Furthermore, it is likely that as a side effect of internal planning failures, new knowledge will be identified that will be helpful, if not critical, to the selection of an appropriate recovery strategy. For example, in (65) above, R's belief about coral's load average not being transmitted to ucbcory might have been made salient as an effect of her failure to find a plan for obtaining coral's load average from ucbcory. In planning her recovery, R could then have decided to use that information to explain to Q why no such plan exists. In laying out the Response Planner's architecture, I have used a feedback channel to accommodate the system's need to react to planning failures.

#### Theoretical problems

The addition of a feedback channel from the Strategic Planner to the Response Goal Proposer—thus allowing the RGP to modify the system's high-level response goals during the planner's operation—leads to difficult theoretical problems. For example, what is to prevent the Response Planner from looping indefinitely, changing its goals each time? What is to keep the system from overgenerating, that is, adding so many goals through feedback that the overall system produces so much output as to violate Grice's maxim of Quantity (see Section 2.1)?

Overgeneration, of course, is a general problem that has never been addressed—and never really can be satisfactorily addressed—by questioner-based systems. When the focus is on reproducing fixed forms of behavior, concerns about prolixity have no place in the research. In fact, in all of the literature that I have studied, I found only two attempts at proposing constraints on cooperative response behavior. One proposal was by Wahlster *et al.* [Wahlster 83], the other by Joshi *et al.* [Joshi 84b]. The first was offered primarily as an implementation technique, the second as a conjecture. Neither proposal has been evaluated in any depth.

Grice's maxim of Quantity reflects the fact that people can and do make judgments about how much information is necessary or appropriate to include in a reply. Our computational models of CRG should be able to do the same. Looking at the problem from the respondent's point of view has the advantage of forcing researchers to consider what overgeneration really means and to investigate principled constraining methods. Of course, this engenders many difficult and challenging questions, and while the framework I am proposing does not make them any less difficult or challenging, it provides a reasonable foundation for recognizing them as problems and confronting them directly.

## 4.4 Concluding Remarks

This chapter has examined the design principles and general structure of a CRG system whose core component is an agent capable of forming and adopting high-level response goals and developing plans to satisfy them. I have called such a system a *Cooperative Response Planning System*, or CRPS. The CRPS was divided into three reasoning components: a Conversation Modeler, a Response Planner, and a Language Generator. I discussed the theory and operation of the first two of these components.

The Conversation Modeler is the component that builds the derived knowledge base—called the *conversation model*—from which the Response Planner initially reasons. The Conversation Modeler is conceived of as comprising only those inference mechanisms that must be routinely invoked prior to response planning.

Initially driven by beliefs found in the conversation model, the Response Planner identifies well motivated high-level response goals for the system to adopt and develop plans to satisfy. Rather than receiving its goals from an external source, it forms them itself. Once the goals are selected, a plan must be found to satisfy them. The adopted goals might not always be achievable, and so there is a need for feedback from the planning component to the goal-selection component.

In the next chapter, I will discuss the processing of two examples of cooperative response behavior within the Cooperative Response Planning System framework developed here.

# Chapter 5

# Analyses of Examples

This chapter analyzes the processing of two examples of cooperative response behavior within the Cooperative Response Planning System framework developed in Chapter 4. The first (considered in Section 5.1) is a COOP-style response [Kaplan 82] generated when the questioner's query is found to presume the truth of a proposition that the system believes to be false. The second example (considered in Section 5.2) is the now-familiar dialogue, taken from the UNIX transcripts, centering on the questioner's goal of sending electronic mail to (some unspecified user or users of) ucbeuler.

## 5.1 Correcting a False Presumption

The following query-response pair exemplifies the kind of exchange that Kaplan's COOP system was developed to handle:

(67) Q: Which COBOL programmers are in department 67?
R: I don't know of any programmers in department 67.

Our interest here is in examining the reasoning process by which a cooperative respondent might first detect and then decide to correct Q's false presumption that there exists a set of programmers in department 67, rather than simply responding with a truthful but misleading answer of "None".

The COOP system demonstrated the feasibility of one technique that database-query applications could employ to provide "cooperative" answers to questions, namely, detecting and correcting false presumptions indicated by extensional query failures. The research was aimed at improving these applications' question-answering competence *as perceived by their users*. As such, it was not necessary to develop more than a relatively superficial analysis of the reasoning processes that might give rise to the desired output behavior. But if we want to develop a *respondent-based* planning model of the behavior that COOP produced, those reasoning processes must be analyzed and understood at a deeper level. I begin by characterizing the conversation model that a respondent could be expected to derive on the basis of Q's utterances (Section 5.1.1) and then consider how the response might be planned using that model (Section 5.1.2).

### 5.1.1 Building the conversation model

Crucial to the evident cooperativeness of the response is R's ability to reason about conflicts between her own beliefs and those she ascribes to Q regarding extensions in the current database of various set descriptions.

#### Basic definitions

I will first define six sets that participate in the reasoning underlying the production of the response in Example (67).[1] I will refer to these sets using the symbols $\mathcal{P}$, $\mathcal{D}$, $\mathcal{C}$, $\mathcal{D}_{67}$, $\mathcal{P}_{\mathcal{D}_{67}}$, and $\mathcal{C}_{\mathcal{D}_{67}}$. The symbol $\mathcal{P}$ will be used to denote the set of all known programmers, $\mathcal{D}$ to denote the set of all known departments:

$$\mathcal{P} = \{x \mid \text{programmer}(x)\}$$
$$\mathcal{D} = \{x \mid \text{department}(x)\}$$

Obviously, we can assume that $\mathcal{P}$ and $\mathcal{D}$ denote disjoint sets. Next, $\mathcal{C}$ denotes the subset of $\mathcal{P}$ of COBOL programmers:

$$\mathcal{C} = \{x \in \mathcal{P} \mid \text{prog\_lang}(x, \text{COBOL})\}$$

$\mathcal{D}_{67}$ denotes the referent of the description "department 67":

$$\mathcal{D}_{67} = \iota x \in \mathcal{D} : \text{dept\_no}(x, 67)$$

In this expression, $\iota x P(x)$ means "the unique $x$ such that $P(x)$". Finally, we define the sets $\mathcal{P}_{\mathcal{D}_{67}}$ of programmers in $\mathcal{D}_{67}$ and $\mathcal{C}_{\mathcal{D}_{67}}$ of COBOL programmers in $\mathcal{D}_{67}$.

$$\mathcal{P}_{\mathcal{D}_{67}} = \{x \in \mathcal{P} \mid \text{in}(x, \mathcal{D}_{67})\}$$
$$\mathcal{C}_{\mathcal{D}_{67}} = \mathcal{C} \cap \mathcal{P}_{\mathcal{D}_{67}}$$

Of particular importance to an understanding of how the response is computed are the set-theoretic rules governing how set emptiness "propagates". For example, since $\mathcal{C} \subset \mathcal{P}$, $\mathcal{C}$ will be empty whenever $\mathcal{P}$ is empty. The relevant axioms are shown below.

$$\mathcal{P} = \emptyset \Rightarrow \mathcal{C} = \emptyset$$
$$\mathcal{P} = \emptyset \Rightarrow \mathcal{P}_{\mathcal{D}_{67}} = \emptyset$$
$$\mathcal{D} = \emptyset \Rightarrow \text{undefined}(\mathcal{D}_{67})$$
$$\text{undefined}(\mathcal{D}_{67}) \Rightarrow \mathcal{P}_{\mathcal{D}_{67}} = \emptyset$$
$$\mathcal{C} = \emptyset \vee \mathcal{P}_{\mathcal{D}_{67}} = \emptyset \Rightarrow \mathcal{C}_{\mathcal{D}_{67}} = \emptyset$$

As we will see shortly, knowledge of these inference rules can be assumed to be shared by Q and R.

#### Elements of the conversation model

The "conversation model" from which COOP worked consisted of these elements:

[1] I will make the simplifying assumption that

$$\forall x[\text{programmer}(x) \Rightarrow \exists y : \text{department}(y) \wedge \text{in}(x, y)]$$

That is, every programmer must be in a department. Thus we need not worry about the sets "programmers in departments" and "COBOL programmers in departments".

- a representation of the query (in the MQL representation language);
- the answer to the query according to the database.

If the query result was zero or "nil", COOP analyzed the query into its constituent sets and checked each one for an empty extension.

A respondent-based account of the reasoning process leading to the response demands a greatly-enriched conversation model. I now consider the elements that should be part of such a model. The respondent should proceed through at least these steps:

1. constructing intensional descriptions of referring expressions used explicitly or implicitly by Q;
2. determining, for each intensional description, whether it is well formed (i.e., whether all constituent relations $xRy$ are valid with respect to the database schema; cf. Mays [Mays 80]);
3. determining, for each well-formed intensional description, its extension as recorded in the current database (if any);
4. identifying the information or action that Q has requested;
5. identifying Q's beliefs about the world as indicated by his query;
6. evaluating Q's beliefs against R's own model of the world.

Note that COOP had no access to beliefs about "higher" goals that Q might be pursuing and therefore did not model Q's plans. Instead, Q's goal was restricted to having R satisfy the request.

**Intensional descriptions** The symbols $\mathcal{P}$, $\mathcal{D}$, $\mathcal{C}$, $\mathcal{D}_{67}$, $\mathcal{P}_{\mathcal{D}_{67}}$, and $\mathcal{C}_{\mathcal{D}_{67}}$ will be used to denote the intensional descriptions that R constructs and uses to represent and reason about Q's query. These symbols will have their set-theoretic definitions.

These descriptions become part of R's conversation model for the exchange. That is, I am suggesting that although the query explicitly mentions only the set of COBOL programmers and the entity "department 67", and implicitly evokes only the set denoted by $\mathcal{C}_{\mathcal{D}_{67}}$, R nevertheless builds and adds to her conversation model the intensional representations corresponding to $\mathcal{P}$, $\mathcal{D}$, and $\mathcal{P}_{\mathcal{D}_{67}}$.[2]

**Checking well-formedness** The next step in constructing the conversation model consists of checking the well-formedness of the intensional descriptions. For the present example, all descriptions will be assumed to be well formed, that is, R's model of the world (the database schema) contains the concepts "programmer", "COBOL programmer", "department", and so forth. I will represent this situation using the following axiom:

[2] I do *not*, however, offer this as a cognitive claim, since I have no data to support it. As I have noted, COOP does not perform such analyses until after a null query result has been obtained. But I will assume that all required set descriptions are built as a routine matter, in order to avoid the details involved in modeling the process by which R might explicitly decide to undertake further analysis of Q's query, alter her response strategy based on the results of that analysis, and so forth.

$\forall i \in \{\mathcal{P},\mathcal{D},\mathcal{C},\mathcal{D}_{67},\mathcal{P}_{\mathcal{D}_{67}},\mathcal{C}_{\mathcal{D}_{67}}\}$ `BMB(R,Q,wf(`$i$`))`

This axiom states that R believes it to be mutually believed between herself and Q that all the referring expressions are well formed.

**Finding extensions** When R tries to determine the extensions of the intensional descriptions she has built and placed in her model of the conversation, she will find a set of programmers, a smaller set of COBOL programmers, a set of departments, and a referent for "department 67". She will not, however, find any programmers in department 67. Consequently, $\mathcal{C}_{\mathcal{D}_{67}}$ will turn out to be empty as well. This will cause the following facts to be added to R's conversation model:

$$\begin{aligned}
&\mathcal{P} = \{\text{p1},\text{p2},\text{p3},\ldots\}\\
&\mathcal{C} = \{\text{p3},\text{p5},\text{p44},\ldots\}\\
&\mathcal{D} = \{\text{d1},\text{d2},\text{d3},\ldots\}\\
&\mathcal{D}_{67} = \text{d67}\\
&\mathcal{P}_{\mathcal{D}_{67}} = \emptyset\\
&\mathcal{C}_{\mathcal{D}_{67}} = \emptyset
\end{aligned}$$

**Representing the query** R's conversation model must also contain some representation of Q's query. As uttered, the query literally requests that R identify the members of $\mathcal{C}$ satisfying the predicate $\lambda x \text{in}(x,\mathcal{D}_{67})$. That set is exactly $\mathcal{C}_{\mathcal{D}_{67}}$, so I will represent the query simply as

`request(Q,R,informref(R,Q,`$\mathcal{C}_{\mathcal{D}_{67}}$`))`

That is, Q has REQUESTed that R INFORM him of the REFerent of the description corresponding to $\mathcal{C}_{\mathcal{D}_{67}}$.[3]

**Modeling Q's beliefs** Next, we consider the beliefs that R ascribes to Q on the basis of the observed utterance (the request). Since Q has asked a question of the form *Which Xs P(X)*, R can reasonably infer that Q believes that there is a set of Xs. That is,

`BEL(Q,not(`$\mathcal{C} = \emptyset$`))`.

More significantly, Q can be expected to believe not only that $\mathcal{C}$ has a non-empty extension, but also that all *generalizations* of $\mathcal{C}$ (cf. Motro [Motro 84]) have non-empty extensions. In particular, R can infer that Q believes $\mathcal{P}$ has a non-empty extension.

`BEL(Q,not(`$\mathcal{P} = \emptyset$`))`.

As I have suggested, Q's query implicitly invokes—and causes R to build representations of—the two sets "programmers in department 67" ($\mathcal{P}_{\mathcal{D}_{67}}$) and "COBOL programmers in department 67" ($\mathcal{C}_{\mathcal{D}_{67}}$). Following Motro once more, we note that while Q need not believe that $\mathcal{C}_{\mathcal{D}_{67}}$ has a non-empty extension, he surely would believe that any *generalization* of that set would be non-empty. In particular,

[3]The operators REQUEST, INFORM, INFORMREF, and so on, are not new, of course. Allen, for example, used them in developing his model of plan recognition [Allen 83], as did Litman in her model of clarification subdialogues [Litman 84b]. Both requesting and informing acts have been formalized by Cohen and Perrault [Cohen 79], and, although the semantics of INFORMREF remain largely unexplored, Appelt has modeled the planned use of definite referring expressions [Appelt 85a].

$\texttt{BEL(Q,not(}\mathcal{P}_{\mathcal{D}_{67}} = \emptyset\texttt{)).}$

This is the belief that drives the respondent's reasoning leading her to provide a corrective response.

**Modeling R's beliefs** Each proposition in the conversation model represents one of the respondent's beliefs. But if we are to develop precise models of cooperative response behavior, we must carefully distinguish at least these kinds of belief:

- R's private beliefs;
- R's beliefs about Q's private beliefs;
- R's beliefs about mutual beliefs.

In the context of Example (67), R has a private belief that $\mathcal{P}_{\mathcal{D}_{67}} = \emptyset$. This belief is *private* in the sense that R does not ascribe this belief to Q. More precisely, we can say the following about R's beliefs:

$\texttt{BEL(R,}\mathcal{P}_{\mathcal{D}_{67}} = \emptyset\texttt{)}$
$\texttt{BEL(R,BEL(Q,not(}\mathcal{P}_{\mathcal{D}_{67}} = \emptyset\texttt{)))}$

These propositions state that R holds the belief $P$ (where $P$ corresponds to "$\mathcal{P}_{\mathcal{D}_{67}} = \emptyset$"), but believes Q to believe not($P$). The second proposition represents one of R's beliefs about Q's private beliefs. Given that there is a disagreement between R's and Q's private beliefs about $P$, we can straightforwardly deduce that $P$ is not believed mutually believed between Q and R:

$\texttt{not(BMB(R,Q,}\mathcal{P}_{\mathcal{D}_{67}} = \emptyset\texttt{))}$

Note, however, that the inference rule

$$\mathcal{P}_{\mathcal{D}_{67}} = \emptyset \Rightarrow \mathcal{C}_{\mathcal{D}_{67}} = \emptyset$$

can be assumed to be mutually believed. That is:

$\texttt{BMB(R,Q,}\mathcal{P}_{\mathcal{D}_{67}} = \emptyset \Rightarrow \mathcal{C}_{\mathcal{D}_{67}} = \emptyset\texttt{)}$

In other words, R should be able to infer that if Q believed $\mathcal{P}_{\mathcal{D}_{67}} = \emptyset$, he would be able to infer $\mathcal{C}_{\mathcal{D}_{67}} = \emptyset$ from that belief.

### Conversation model for COOP example

We now have enough conceptual machinery to state the conversation model from which R will reason to plan a cooperative response. The complete model is shown in Figure 5.1. The model is divided into several parts; these are briefly summarized next.

**Definitions of intensional descriptions** The intensional descriptions are "definitional" in the sense that they define sets referred to explicitly in or evoked by Q's query. They must be part of the conversation model because other beliefs about the conversation are defined in terms of them. In effect, they represent the respondent's beliefs about sets of entities to which the questioner intends to refer.

Definitions of intensional descriptions:

$\mathcal{P} = \{x \mid \text{programmer}(x)\}$
$\mathcal{D} = \{x \mid \text{department}(x)\}$
$\mathcal{C} = \{x \in \mathcal{P} \mid \text{prog\_lang}(x, \text{COBOL})\}$
$\mathcal{D}_{67} = \iota x \in \mathcal{D} : \text{dept\_no}(x, 67)$
$\mathcal{P}_{\mathcal{D}_{67}} = \{x \in \mathcal{P} \mid \text{in}(x, \mathcal{D}_{67})\}$
$\mathcal{C}_{\mathcal{D}_{67}} = \mathcal{C} \cap \mathcal{P}_{\mathcal{D}_{67}}$

Inference rules about set relationships:

BMB(R,Q,$\mathcal{P} = \emptyset \Rightarrow \mathcal{C} = \emptyset$)
BMB(R,Q,$\mathcal{D} = \emptyset \Rightarrow \text{undefined}(\mathcal{D}_{67})$)
BMB(R,Q,$\text{undefined}(\mathcal{D}_{67}) \Rightarrow \mathcal{P}_{\mathcal{D}_{67}} = \emptyset$)
BMB(R,Q,$\mathcal{P}_{\mathcal{D}_{67}} = \emptyset \Rightarrow \mathcal{C}_{\mathcal{D}_{67}} = \emptyset$)
BMB(R,Q,$\mathcal{C} = \emptyset \Rightarrow \mathcal{C}_{\mathcal{D}_{67}} = \emptyset$)

Well-formedness beliefs:

$\forall i \in \{\mathcal{P}, \mathcal{D}, \mathcal{C}, \mathcal{D}_{67}, \mathcal{P}_{\mathcal{D}_{67}}, \mathcal{C}_{\mathcal{D}_{67}}\}$ BMB(R,Q,wf($i$))

Model of Q's request:

BMB(R,Q,request(Q,R,informref(R,Q,$\mathcal{C}_{\mathcal{D}_{67}}$)))

Beliefs ascribed to Q:

BEL(Q,not($\mathcal{C} = \emptyset$))
BEL(Q,not($\mathcal{P} = \emptyset$))
BEL(Q,not($\mathcal{P}_{\mathcal{D}_{67}} = \emptyset$))
BEL(Q,not($\mathcal{C}_{\mathcal{D}_{67}} = \emptyset$))

R's private beliefs:

$\mathcal{P} = \{\text{p1}, \text{p2}, \text{p3}, \ldots\}$
$\mathcal{C} = \{\text{p3}, \text{p5}, \text{p44}, \ldots\}$
$\mathcal{D} = \{\text{d1}, \text{d2}, \text{d3}, \ldots\}$
$\mathcal{D}_{67} = \text{d67}$
$\mathcal{P}_{\mathcal{D}_{67}} = \emptyset$
$\mathcal{C}_{\mathcal{D}_{67}} = \emptyset$
not(BMB(R,Q,$\mathcal{P}_{\mathcal{D}_{67}} = \emptyset$))
not(BMB(R,Q,$\mathcal{C}_{\mathcal{D}_{67}} = \emptyset$))

Figure 5.1: R's Conversation Model for COOP Example

**Inference rules about set relationships** For this example, we can assume that Q and R share all (set-theoretic) beliefs about how emptiness of various sets "propagates". If, for example, R did not believe that Q could infer the emptiness of the set of COBOL programmers from belief of the emptiness of the set of programmers, she might be inclined to augment her response with an appropriate explanation or justification.

**Well-formedness beliefs** For the response to be appropriate, we must assume that both dialogue participants believe all referring expressions to be well formed.[4] As Mays showed, ill-formed intensional descriptions may demand special response strategies [Mays 80]. Therefore, we require that R's beliefs about well formedness be part of her model of the conversation.

**Model of request** Both the questioner and the respondent can be assumed to believe that a request has taken place, where the object of the request is an act by R of informing Q of the referent of the description "COBOL programmers in department 67". If the conversation model were not to reveal any other conditions for R to address in her response, she could use her belief about the request to plan a response that would satisfy it.

**Beliefs ascribed to Q** While many beliefs might be inferable from Q's query, at least four are crucial. First, Q believes that there is a non-empty set of COBOL programmers (in the world represented by the database). Similarly, he also believes there is a non-empty set of programmers. Third, Q believes that restricting the (non-empty) set of programmers to those in department 67 yields a non-empty set. He has a similar belief regarding the restriction of the set of COBOL programmers to those in department 67.

**R's private beliefs** The respondent, having access to the knowledge recorded in the database, is able to determine the extensions of the intensionally-defined sets. Through this process she acquires particular beliefs, including, for example, the belief that the set $\{\mathrm{p1}, \mathrm{p2}, \mathrm{p3}, \ldots\}$ is the extension of the set represented by $\mathcal{P}$. Of particular importance here is the belief that R acquires when she restricts $\mathcal{P}$ to those programmers in department 67. An empty set results, which leads to the emptiness of the set denoted by $\mathcal{C}_{\mathcal{D}_{67}}$. These two beliefs disagree with related beliefs ascribed to Q, so R can infer that her particular beliefs about $\mathcal{P}_{\mathcal{D}_{67}}$ and $\mathcal{C}_{\mathcal{D}_{67}}$ are not shared by Q.

### 5.1.2 Response goals proposed and adopted

Much work remains to be done, of course, before we can truly say that we have developed a complete and precise specification of the conversation model that R builds on the basis of Q's query in Example (67). In particular, more attention needs to be paid to questions concerning which beliefs are private and which are believed mutually believed. Figure 5.1 is only a first approximation to such a model. Nevertheless, the model is instructive in that it provides insight into the kinds of rich knowledge bases that might be needed to support cooperative response planning.

[4] It would, however, be interesting to consider the consequences of weakening this assumption from one-sided mutual belief to a private belief of R.

Developing a model of the respondent's goals in producing the response shown in the example demands that we address these two questions:

1. What principles motivate the decision to offer a corrective response?
2. What principles explain R's decision to omit the correct, albeit misleading, direct answer?

Adopting the respondent-based perspective means acknowledging the relevance of these questions. Finding satisfactory answers to them is an open research issue; for the present, I will discuss some proposals.

The decision to provide a corrective response must be related to the respondent's reasoning about the disagreement she has discovered between her own beliefs and those she has ascribed to Q. Given our view of the conversation model as circumscribing all those beliefs that are believed relevant to the current conversation, perhaps we might be able to use the principle of "squaring away" mutual beliefs as discussed by Joshi [Joshi 82] to motivate the corrective response behavior. We might reformulate that principle in the following way:

*Maximize mutuality of beliefs in the conversation model.*

This principle reflects the theory that at least one goal of cooperative interaction might be to reach agreement on all beliefs that have been stated or implied over the course of a dialogue.

Bearing this principle in mind, we note that the conversation model proposed for the present example shows that there are two propositions over which Q's and R's beliefs differ (the emptiness or non-emptiness of the sets $\mathcal{P}_{\mathcal{D}_{67}}$ and $\mathcal{C}_{\mathcal{D}_{67}}$). This state of affairs might lead R to adopt the following high-level goals:

$$\texttt{BMB(R,Q,}\mathcal{P}_{\mathcal{D}_{67}} = \emptyset\texttt{)}$$
$$\texttt{BMB(R,Q,}\mathcal{C}_{\mathcal{D}_{67}} = \emptyset\texttt{)}$$

(Note that I am making what Pollack has labeled the "correct knowledge assumption" [Pollack 86], that is, I am assuming that R's beliefs are always correct.) In other words, the principle of maximizing mutuality of belief in this case motivates R to adopt the goals of causing her private beliefs about the sets $\mathcal{P}_{\mathcal{D}_{67}}$ and $\mathcal{C}_{\mathcal{D}_{67}}$ to become mutually believed between herself and Q. These, then, are at least two of the goals that the Response Goal Proposer might add to the Response Goals blackboard.

Devising a principled account of the omission of the direct answer is considerably more difficult. One way to look at it might be to have the respondent adopt a "satisfy request" goal, and then later decide to drop it. Deciding to drop such a goal would entail reasoning about the possible effects on the questioner of providing a direct answer. For example, R might have to determine that a reply of "None" would have the effect of permitting Q to maintain his (incorrect) belief that the set $\mathcal{P}_{\mathcal{D}_{67}}$ has a non-empty extension. This is a non-trivial reasoning process. On the other hand, perhaps a better way to view the response is not as *omitting* the direct answer, but rather as *providing it implicitly.* That is, R might be able to reason that Q can infer the direct answer from her assertion that the set $\mathcal{P}_{\mathcal{D}_{67}}$ is empty.

Both accounts entail a complicated reasoning process wherein the respondent (1) chooses a set of response goals, (2) plans utterances to achieve those goals, (3) models the effects of those utterances on (her model of) the questioner's beliefs, and then (4) reasons about the results, possibly iterating through these steps several times, changing or modifying her response plan on each iteration. This seems to be the correct approach in the long run, but there are many technical details that remain to be worked out.

### 5.1.3 Strategic planning

For the sake of discussing strategic planning, we will take R's adopted goals to be:

$\texttt{BMB(R,Q,}\mathcal{P}_{\mathcal{D}_{67}} = \emptyset\texttt{)}$
$\texttt{BMB(R,Q,}\mathcal{C}_{\mathcal{D}_{67}} = \emptyset\texttt{)}$

Looking at the response actually provided in the example, it appears that R's surface goal is:

$\texttt{inform(R,Q,not(BEL(R,not(}\mathcal{P}_{\mathcal{D}_{67}} = \emptyset\texttt{))))}$

That is, R's surface goal is, roughly, to inform Q that R does not believe that the set of programmers in department 67 is non-empty.

According to the theory developed in Chapter 4, the Strategic Planner is responsible for computing the plan that links the respondent's adopted high-level response goals to the surface goals. In the context of the present example, this raises the question of how exactly an agent might plan to cause a proposition $P$ to become mutually believed. For simple propositions like those underlying the response in Example (67), it is probably safe to assume that an assertion of $P$ by R will cause $P$ to become mutually believed. This is reasonable because:

- It will be mutually believed that R is sincere.
- It will be mutually believed that R has correct knowledge.

Given that Q will have no reason to believe either that R is dissembling or is herself misinformed, R can assume that her assertions will lead to mutual belief. Even so, the two goals proposed above lead to the question of why R does not then respond as in (68).

(68) R: I don't know of any programmers in department 67. I don't know of any COBOL programmers in department 67.

Clearly, the second proposition logically follows from the first. The corresponding inference rule is, in fact, part of the conversation model. This suggests that another principle is needed. Consider the following proposal:

*Make explicit in a response only those propositions that the hearer cannot easily infer.*

The point here is that R believes that Q can deduce the emptiness of the set of COBOL programmers from knowledge about the emptiness of the set of programmers in general. R should be able to exploit this knowledge in order to produce a concise response.

Of course, the fact that we have had to propose a general principle in order to understand the Strategic Planner's operation suggests that there are principles of cooperation operating at all levels of the response-planning process. The architecture design may thus provide assistance in identifying different classes of rules of cooperative dialogue.

### 5.1.4 Summary

In this section I sketched the steps involved in computing the response shown in Example (67), and considered some of the issues that will have to be resolved before the model can be implemented. In developing a conversation model for the example, we saw that even for such a "simple" example, in a highly-restricted database-query domain, the kind of conversation model that a principled account of the response demands the respondent to possess will need to be very rich indeed.

Two principles were proposed in order to motivate goals that the respondent could plausibly have adopted:

- *Maximize mutuality of beliefs in the conversation model.*
- *Make explicit in a response only those propositions that the hearer cannot easily infer.*

It remains an open question as to how such principles could be declaratively encoded in a cooperative response planning system and used to determine the system's behavior.

## 5.2 Sending Electronic Mail

As a second exploration of response planning in the theoretical framework developed in this dissertation, I will sketch the processing of the following example (seen several times earlier) culled from the UNIX transcripts:

(69) Q: Do you know if there is any network link to ucbeuler? I want to send some mail over there.
R: The network link is not up yet. Some time in the near future it will be connected to the Berknet, but I don't know exactly when. It is mostly a software change on ucbvax.

The reasoning underlying the response in Example (69) is significantly different from that underlying Example (67). This is primarily because COOP had no access to, and thus could not reason about, Q's goals and plans in the world, whereas reasoning about Q's goals and plans in the world is an important component of any computational model of R's reply above.

Section 5.2.1 examines the conversation model that R must build before planning her response. Section 5.2.2 discusses the high-level response goals that are proposed and adopted, and Section 5.2.3 considers the process of selecting surface goals to achieve the adopted response goals. The results of this investigation are summarized in Section 5.2.4.

### 5.2.1 Building the conversation model

I begin by outlining the conversation model that R can reasonably be expected to build on the basis of Q's utterances. This model will contain at least:

- a representation of Q's explicit request;
- a representation of Q's explicitly-stated goal;

- a representation of Q's domain plan, which relates his request to his goal;
- an evaluation of that plan.

These elements are discussed in the following subsections.

### Modeling the request and goal

I will model Q's first utterance ("Do you... to ucbeuler?") as an explicit request, and treat his second utterance ("I want...over there") as an explicit statement of the *most general goal* to which Q intends the conversation to be related. That is, I am suggesting that Q utters "I want to send some mail over [to ucbeuler]" so as to provide R with a clearly-defined goal on which Q would like the conversation to focus. Consider, for example, how different the response would be if the second utterance were to be replaced by any of (70)a–c.

(70) a. "I want to transfer some files from there."
b. "I want to see if my friend Joe is logged on there."
c. "My friend Joe has been having trouble sending e-mail to me from there."

Each of the above substitute utterances would serve to identify a different reason—and thus lead R to infer a different domain plan—that motivates Q's making of the request.

Thus R's model of the conversation surely must contain a representation of both Q's request and his goal statement. I will use the following notation for these elements:

$$\begin{array}{l} \texttt{request(Q,R,} \\ \qquad \texttt{informif(R,Q,}\exists x\texttt{:netwk\_link} \\ \qquad\qquad\qquad \texttt{connects(}x\texttt{,}\mathit{current\_machine}\texttt{,ucbeuler)))} \\ \texttt{goal(Q,do(Q,send\_mail(}\mathit{current\_machine}\texttt{,ucbeuler)))} \end{array}$$

Q's first utterance is modeled as a request from Q to R that R perform an act of informing Q of the truth status of the proposition "there exists a network link between the current machine and ucbeuler", where the "current machine" refers to the machine on which the dialogue is being held. (Note that to reduce the complexity of the analysis I am ignoring the indirect nature of the request; see, for example, Allen [Allen 83], Allen and Hinkelman [Allen 89], and Hinkelman [Hinkelman 90] for work on the problem of understanding indirect requests.) Q's goal will be modeled simply as the performance of an act of sending mail from the current machine to (some unspecified user or users of) ucbeuler.

### Plan recognition

Once R's model of the conversation has been initialized with the representations of Q's request and goal, her next processing step is to apply a plan-recognition procedure to that model in order to identify the underlying domain plan motivating Q's two utterances. In a sense, plan recognition is used to generate an "explanation", in terms of goals and plans, of the relationship between Q's request and his goal statement. Through this process R should acquire at least these beliefs:

- Q's goal is to send (or be able to send) electronic mail to (some unspecified user or users of) ucbeuler.

- Q has formed a plan to achieve that goal and is considering executing it.[5]
- Q believes that a network link must exist between the current machine and ucbeuler in order for the plan to be executable.
- Q does not know whether such a link exists, and thus does not know whether his plan is executable.
- Q believes that R knows (or at least might know) whether such a link exists.
- Q has made his request of R in order to determine whether his plan is executable.

The most significant result of this process is that R acquires the belief (noted in the third item above) that Q believes the existence of a network link between the current machine and ucbeuler is an *applicability condition* on his plan for sending mail to ucbeuler. This state of affairs could be represented using the following notation:

$$\texttt{acond}(\exists x\texttt{:}\ \ \texttt{netwk\_link connects}(x\texttt{,}\mathit{current\_machine}\texttt{,ucbeuler}),\\ \texttt{do}(\texttt{Q,send\_mail}(\mathit{current\_machine}\texttt{,ucbeuler})))$$

As used in the literature (see, for example, Litman and Allen [Litman 84a]), applicability conditions are like *preconditions* in that they are conditions that must be satisfied in order for an action to be executable. They differ from preconditions, however, in that they are conditions over which the planning agent (typically) has no control; thus the planner should not try to form subplans to change the status of applicability conditions. Because the condition "a network link exists between the current machine and ucbeuler" is one that users typically cannot affect, I model it here as an applicability condition rather than as a simple precondition.

### Plan evaluation

Once a plan has been inferred, R must *evaluate* that plan with respect to her own beliefs. As Pollack argued [Pollack 86], this involves, among others, the following sorts of activities:

- for each precondition (or applicability condition) $P$ of each action $A$ in the plan, checking whether R believes that $P$ is actually a precondition (or applicability condition) of $A$;
- for each precondition (or applicability condition) $P$ of each action $A$ in the plan, checking whether R believes that $P$ will hold at the intended execution time of $A$;
- for each effect $E$ of each action $A$ in the plan, checking whether R believes that $E$ is actually an effect of $A$.

Judging from the information conveyed in the response, we can infer that R believed Q's plan to be completely well formed, but unexecutable. That is, R determined that Q's plan was a perfectly reasonable one, but simply could not be executed because one of its applicability conditions (the existence of a network link) was not satisfied. Consequently, her response focuses on the status and pending establishment of a network link to ucbeuler.

Consider how the response might differ if R's evaluation of Q's plan were different:

[5] See Ramshaw [Ramshaw 91] for related work on a theory of dialogue that permits dialogue systems to distinguish between plans that are actually intended and those that are merely under consideration.

(71) R: Just do this: mail ruby!euler!user. Let us know if it doesn't work. Euler is only reached thru the ruby machine.

The response in (71), a response from the UNIX transcripts that was provided by a different respondent to the query in (69), focuses on describing a method Q can use to achieve his **send_mail** goal, rather than on the existence of a network link. This behavior could be explained by assuming that in this second case R determined that Q's plan was not only well formed but also executable. In other words, R concluded that both Q's plan and query were reasonable. However, once those facts were established, R was able to reason that the answer to the query would be rendered irrelevant to Q's stated goal if she could instead provide information that would directly enable Q to carry out his plan. Providing the necessary UNIX command line serves that goal, hence the response.

In modeling the planning of the response shown in Example (69), I will assume that the following beliefs, generated by the plan-evaluation mechanism, are added to R's conversation model:

- Q's plan is well formed.
- The applicability condition "a network link exists between the current machine and ucbeuler" on Q's plan will not hold at the intended time of execution of the plan.
- Due to the failure of one of its applicability conditions, the plan will be unexecutable at its intended execution time.

We now have enough knowledge in the conversation model to consider how the response-planning process might proceed. Figure 5.2 summarizes the model.

### 5.2.2 Response goals proposed and adopted

If we analyze the response into the propositions that it conveys, we get a set roughly like the following:

- "A network link between the current machine and ucbeuler does not exist at the present time."
- "A network link between the current machine and ucbeuler will be established at some time in the near future."
- "[R does] not know a more precise specification (than *in the near future*) of the time at which the network link to ucbeuler will be established."
- "Establishing the network link primarily involves making a change to software on ucbvax."

A respondent-based model of the response should provide a principled account of the reasoning that connects the beliefs contained in R's model of the conversation to the propositions conveyed in the response's constituent utterances. I discuss the general outline of such an account here.

Model of Q's request:

```
request(Q,R,
        informif(R,Q,
                 ∃x: netwk_link
                     connects(x,current_machine,ucbeuler)))
```

Model of Q's goal:

```
goal(Q,do(Q,send_mail(current_machine,ucbeuler)))
```

R's private beliefs (derived using plan recognition):

Q's goal is to send (or be able to send) electronic mail to (some unspecified user or users of) ucbeuler.

Q has formed a plan to achieve that goal and is considering executing it.

Q believes that a network link must exist between the current machine and ucbeuler in order for the plan to be executable.

Q does not know whether such a link exists, and thus does not know whether his plan is executable.

Q believes that R might know whether such a link exists.

Q has made his request of R in order to determine whether his plan is executable.

R's private beliefs (derived using plan evaluation):

Q's plan is well formed.

The applicability condition "a network link exists between the current machine and ucbeuler"on Q's plan will not hold at the intended time of execution of the plan.

Due to the failure of one of its applicability conditions, the plan will be unexecutable at its intended execution time.

Figure 5.2: R's Conversation Model for UNIX Example

**Satisfying the request** On the basis of the beliefs in R's conversation model, it is reasonable to expect the respondent to initially adopt a high-level response goal of satisfying Q's explicit request. This is because the conversation model contains the following beliefs about Q:

- Q has a well-formed plan under consideration.
- In order to decide whether or not to put the plan into execution, Q needs to know whether the proposition "a network link exists between the current machine and ucbeuler" holds.
- Determining the truth value of that proposition is Q's reason for making the request.

We can think of these beliefs as together providing evidence that Q's request is a "reasonable" one, i.e., a request for which R can find no reason to choose not to satisfy it. (Thus we may want to consider whether "satisfying reasonable requests" should be an axiom of a theory of cooperative response generation.)

In order to satisfy the request, the respondent must determine *for herself* whether she believes a network link exists. That is, she must adopt a (sub) goal of checking whether she believes that the proposition "there exists a network link between the current machine and ucbeuler" is satisfied. Judging from the response, we can infer that R acquired at least the following beliefs through this process:

1. R does not believe that the proposition "there exists a network link between the current machine and ucbeuler" holds at the time of the dialogue.
2. R believes that the truth value of the proposition "there exists a network link between the current machine and ucbeuler" will change from FALSE to TRUE at some time point $t$ which falls within the temporal interval describable as *the near future*.

These constitute new beliefs that are added to R's conversation model as a side effect of her efforts to acquire the information she needs to carry out her response goal of satisfying Q's explicit request. Note that now the conversation model contains information that was not available when the respondent originally began selecting response goals. What is of interest is how this modification affects the respondent's subsequent response planning.

**Adding relevant information** The bulk of the response focuses on the time at which R believes a network link will be established. This raises the question of what general principle (or principles) might be driving this behavior. One hypothesis is that respondents always try to provide *some* information related to the questioner's goal in their responses if they cannot actually help directly. In Example (69), R knows that she cannot offer any information that will immediately help Q achieve his goal. She believes, however, that Q's goal will be achievable relatively soon. She judges that knowing this might be of interest to Q, and since it is the best she can offer, she decides to convey her belief to Q that the network link will be established "in the near future". (Thus we see the importance of reasoning about the questioner's goals.)

In summary, a reasonable step-by-step account of the respondent's goal-selection process is as follows:

1. Based on her beliefs as recorded in the initial conversation model, R adopts the goal of satisfying Q's explicit request.

2. As a step towards achieving that goal, R adopts a goal of determining whether she herself believes that a network link exists between the current machine and ucbeuler.

3. R carries out the reasoning necessary to achieve that goal, discovering in the process that she believes (1) there is no network link at present, and (2) a network link will be established soon. Her conversation model is updated accordingly.

4. Reasoning over this new conversation model, R recognizes that she can satisfy Q's request, but that this would leave his domain goal (of being able to send mail to ucbeuler) unsatisfied.

5. In an effort to at least provide some information judged relevant to Q's domain goal, R adopts another response goal of conveying to Q her beliefs about the time at which a network link will be established.

Thus the two goals ultimately provided to the Strategic Planner are (1) satisfy Q's explicit request, and (2) inform Q when a network link between the current machine and ucbeuler will be established.

### 5.2.3 Strategic planning

Once the planner's goals have been established by the Response Goal Proposer, the Strategic Planner must compute a plan for achieving them using utterance acts.

Developing a plan to satisfy the first goal might at first glance appear to be relatively straightforward. But the response shown in the example suggests it is actually a rather complex process: note the use of the temporal adverbial "yet". This indicates that the plan to satisfy the first goal is influenced by the system's beliefs about the temporal nature of the existence of the network link. Rather than planning to use a simple declarative INFORM utterance act, e.g., "The network link is not up" or "There is no network link", it would seem that the system must instead be able to reason about how the performance of such an act would support or undermine other goals that it would like to achieve. It could, of course, select such an act, but this would most likely demand the later use of constructions like "However..." in order to block the questioner's possible conclusion that the situation is not going to change. This, however, takes us beyond issues of cooperation and into the domain of planning for natural language generation.

The two propositions conveyed by the next utterance— "A network link to ucbeuler will be established in the near future" and "I do not know a more precise specification (than *in the near future*) of the time at which the network link to ucbeuler will be established"— can be treated as being offered in an attempt to "approximately" satisfy the (second) goal of informing Q *when* the link will be established. What is interesting here is that R has adopted a goal that she cannot fully satisfy, due to insufficient knowledge.

Finally, the question arises concerning how we might account for R's concluding utterance: "It is mostly a software change on ucbvax". This utterance has nothing to do with satisfying the request, nor does it directly follow from a goal of informing Q when a network link will be established. It seems to be offered in a spirit of suggesting that not much work is needed to establish the link (it is "only a matter of a software change") and so Q's goal is likely to be achievable sooner rather than later.

One possible approach to modeling this behavior would be to take advantage of the architecture's capacity for reflection. For example, once the Strategic Planner has formed its plan to (1) satisfy Q's request, and (2) communicate the system's available knowledge about the time at which a network link will be established, this plan could be added to the conversation model, representing an intermediate stage of R's planned response. This could allow the Response Goal Proposer to reason about that plan and generate new goals. In this example, the RGP might reason that, if the system carries out its current response plan, Q might then be interested in any information concerning factors that might accelerate or inhibit establishment of the network link. Although there is good reason to believe that such forms of reasoning will be necessary to account for such examples, it remains a challenging problem to find a satisfactory computational model of this process.

### 5.2.4 Summary

This section sketched a model of the processing of a reasonably complicated naturally occurring example culled from the UNIX transcripts. We saw that building a conversation model demanded the invocation of mechanisms for plan recognition and for plan evaluation. Although the response-planning process was described only in general terms, the exercise nevertheless provided further insight into the kinds of issues that respondent-based modeling of cooperative response phenomena forces theorists to address.

## 5.3 Concluding remarks

This chapter outlined respondent-based models of two examples of cooperative response behavior. In the first example, we saw how taking the respondent's perspective in cooperative response generation led to the specification of a rich and detailed model of the respondent's beliefs prior to her response-planning activity. This model then served as a starting point for stating and evaluating principles of cooperative behavior that might be governing the response. The analysis of the second example, a naturally occurring exchange in the richer domain of UNIX advising, demonstrated the significance of both plan inference and evaluation knowledge in the respondent's model of the conversation. Furthermore, we saw how a detailed account of response planning might require that respondents be able to adopt, pursue, and later revise their high-level response goals.

# Chapter 6

# Conclusions and Directions for Further Work

This dissertation has examined research and design issues pertaining to the development of natural-language dialogue systems intended to be able to provide cooperative responses. In this chapter, I will discuss both the broad conclusions that can be drawn from this work and the specific results that have been obtained, and will outline some interesting avenues for further exploration.

## 6.1 Thesis Contributions

The research I have presented herein constitutes a top-down re-evaluation of efforts to develop computational models of cooperative response behavior. The work began as an effort in the engineering of natural-language dialogue systems. After over a decade of work by others on cooperative question answering, it seemed logical to examine the problem of extending and combining—or *integrating*—those efforts into a single, more generally capable cooperative dialogue system. But difficulties encountered during this effort suggested that individual projects, while identifying interesting regularities in human cooperative response behavior, had made assumptions about their domains and tasks that were sufficiently dissimilar as to call their integrability into question.

### 6.1.1 General conclusions

The re-evaluation that was started in response to these problems yielded the first conclusion of the present work, namely, that without scientific advances in understanding the nature of cooperation and cooperative response behavior, continued work on engineering solutions will not lead us to "truly cooperative" dialogue systems.

Building on past work, studying both the methods that have been used and the results that were obtained, I identified a paradigm that captures the essence of previous accounts of cooperative responses. I found that researchers had been developing models that implicitly adopted the questioner's viewpoint in their conceptions of the notion of "cooperativeness". In any given system, the definition of what actually made its responses "cooperative" was either left to intuition or stated in terms of some desired effect on the questioner (e.g., preventing him from drawing an incorrect conclusion from the system's answer). Consequently, I termed this research paradigm the *questioner-based perspective*.

I developed a detailed characterization and critique of the questioner-based perspective and showed that, when cooperative responses are modeled in that theoretical framework, it naturally follows that the abstract principles driving a system's behavior can be overlooked. Furthermore, the questioner-based approach makes it possible to model different responses—and even different kinds of responses to the same query—in non-uniform ways, directly impeding integration efforts. I concluded that the questioner-based paradigm ought to be abandoned since it supports neither scientific nor engineering progress.

The insights resulting both from the characterization of the questioner-based perspective and from numerous analyses of naturally-occurring request-response pairs gave rise to the formulation of a new research paradigm, which I have called the *respondent-based perspective*. According to this paradigm, responses are viewed as the observable result of the respondent's plan to achieve one or more goals. It demands that cooperative responses be modeled in terms of, e.g., the general reasoning principles that drive the respondent's behavior. Most importantly, the respondent-based paradigm holds that it is the *process of cooperative reasoning*, rather than the manifestations of that process, that should to be the object of study.

Using examples drawn from transcripts of naturally-occurring dialogue, I discussed several dimensions of cooperative response behavior, such as knowledge-dependent response variations and notions of responsibility and willingness to cooperate, that cannot be properly accounted for within the questioner-based paradigm. Modeling the respondent's reasoning process, however, should make it possible to accommodate these ideas.

Adopting the respondent-based perspective focuses attention directly on the scientific problem: What are the principles of a theory of cooperation and cooperative dialogue? Research along these lines promises to lay the foundation for real engineering progress toward integrated cooperative dialogue systems, and so I conclude that it is time for a paradigm shift from the questioner's to the respondent's perspective.

Carrying through such a shift poses many difficult problems. In the third major portion of this dissertation (Chapter 4), I used analyses of naturally-occurring data to motivate the general structure of a planning system for respondent-based cooperative response generation. I outlined the system's top-level organization, and then explored the designs of its two most important processing elements. The resulting architecture is useful primarily for the way it partitions what has been a poorly-understood general reasoning task. In going through the exercise of developing the architecture, several significant unsolved theoretical problems have been identified, thus pointing the way to future progress. Some of these problems will be summarized in Section 6.2.

### 6.1.2 Specific results

Besides the general conclusions presented above, my investigations into the problem of generating cooperative responses have produced several specific results.

One such result is my identification of five elements of a respondent-based model of CRG. I distinguished:

1. the goals that respondents adopt and try to achieve in their responses;

2. the conditions of the conversation that motivate various response goals;
3. the knowledge and reasoning resources needed to determine the status of the goal-motivating conditions;
4. the reasoning processes and principles by which response goals are actually adopted;
5. the strategies that may be used to achieve a given response goal.

Research effort spent on the first, second, and fourth elements should directly advance our comprehension of the principles of cooperative interaction. Those principles will define the kinds of tests that respondents apply to the representations they derive from questioners' utterances (the second element), the connections between those tests and the high-level goals whose adoption they motivate (the first element), and the more general processes of deciding when well-motivated goals should be pursued in a given response situation (the fourth element). Knowing the kinds of conversational conditions that influence response-goal decisions will provide grounding both for the design of computational mechanisms that produce the supporting data, like plan inference procedures, and for the actual tests on that data, like obstacle detection and belief checking. Once we understand the kinds of high-level goals that cooperative respondents adopt, analysis of naturally-occurring data should provide clearer insight into the methods people employ to achieve those goals.

These are elements that participate in a general reasoning process, and the architectural model discussed in Chapter 4 represents a first attempt at characterizing their roles in that process. I partitioned that process into three subtasks: a stage of building a model of the conversation, a stage of selecting response goals and developing a plan to satisfy them, and a language generation stage.

In developing the structure of the planning component of a CRG system, I explored two important architectural properties. First, the system must be able to formulate its own goals, rather than receiving them from an external source. This process breaks down into a phase of identifying well-motivated goals and then applying more general "screens of admissibility" to select a subset of those goals to adopt and pursue. Second, there is a need for some kind of internal feedback loop in the planning stage. Analysis of naturally-occurring data suggests both that the respondent may adopt high-level goals that are later found to be unachievable and that during the process of computing a plan to satisfy one or more high-level goals, new information may become available that motivates the adoption of new high-level goals.

In carrying out the various analyses upon which much of this dissertation work has been based, I discovered the importance of comparative study of cooperative response data. Past work on cooperative response generation has usually been restricted to simulating some small set of (often hand constructed) examples. But just as one needs many different examples of a word's use to ascertain the range of its senses, it is essential to consider many different examples of cooperative response behavior to gain insight into the principles of reasoning that might underlie cooperative discourse. The UNIX transcripts were particularly useful in this regard in that they contained several examples in which different respondents provided different responses to the same request. These examples create an important challenge for investigators of cooperative phenomena: to develop compatible models that account for such response variations.

## 6.2 Some Open Questions

This section summarizes the major open questions that are raised by the Cooperative Response Planning System (CRPS) architecture I outlined in Chapter 4.

### Identifying response goals

Bearing in mind the five elements of a respondent-based CRG model, I characterized the Response Goal Proposer (RGP) as applying tests against the conversation model to identify high-level response goals that are well motivated in the response situation in focus.

It has never been clear, however, exactly how to identify a principled set of such high-level goals. To the extent that UCEGO reasoned from high-level goals, Chin simply chose a set that seemed reasonable in the context of the domain and that fit the needs of the responses he wanted the system to generate [Chin 88a].

It seems clear that if we restrict our attention only to small sets of hand-picked examples, it will be easy to define high-level goals in *ad hoc* ways. Large corpora, such as those I have examined over the course of this work, provide much better insight into the underlying principles of behavior, but they are difficult to deal with, identify patterns in, and extract solid conclusions from. That is perhaps what makes the respondent-based approach tricky: the only reliable source of insight into cooperative discourse is naturally-occurring data, but the standard approach of searching for and modeling regular patterns of behavior tends to lead one away from those insights.

### Proposing and adopting response goals

I have argued that it makes sense to separate the task of identifying the high-level response goals that are well motivated based on the conversation model from the process of reasoning about the resulting set and choosing some subset of goals to actually adopt. I suggested that the RGP might have access to a library of abstract high-level goals, where each element of that library would have a set of *activation conditions* associated with it. When a particular goal's activation conditions were satisfied (according to tests against the conversation model), it would be instantiated and added to the RGP's internal list of potential response goals. When all such goals had been identified, some set of *screens of admissibility* would be employed to select the goals for the system to adopt.

Although there are arguments to support the distinction between activation conditions and screens of admissibility, it is not clear what methods could be used to define them, especially the latter. Activation conditions, at least, might be abstracted from previous research. Screens of admissibility, however, were proposed in acknowledgment of the fact that realistic response planning systems will likely have to reason about multiple goals. Since such possibilities were never allowed in the past, no one ever considered the idea that goals that were well motivated individually might have negative interactions with each other or might conflict with other goals the respondent could hold.

Ultimately, of course, the question that will have to be answered is this: What is the connection between activation conditions, screens of admissibility, and principles of cooperation?

**Interaction of RGP and SP**

There is evidence to support the hypothesis that some kind of feedback connection must exist between the component that suggests goals for the system to pursue and the component that attempts to compute plans to satisfy them. But the properties of that feedback process are not well understood. For example, what information does the SP send back to the RGP? Could the RGP ever decide to discard its original set of adopted goals and propose a new set? How would that affect the SP's design? Many issues with respect to the RGP/SP interface need to be investigated.

## 6.3 Some Directions for Future Research

During my investigations of general cooperative response phenomena, a few more narrowly-focused research questions were identified which might serve as interesting directions for future study. I sketch them here.

### 6.3.1 A theory of question asking

It is worth noting that no computational theory yet exists that explains how the *questioner* selects the two utterances (72)a and (72)b (the example is taken from Pollack's dissertation [Pollack 86]).

(72)a. Q: I want to talk to Kathy.
    b. What's the phone number of the hospital?
(73)a. R: Kathy has been discharged
    b. and is at home now.
    c. The phone number there is 222-1234.

Related research includes some early work on the planning of requests [Cohen 78, Cohen 79], more recent work on the planning of referring expressions [Appelt 85a], and some work on phrasing questions understandably in the context of cooperative problem-solving tasks [Webber 84]. However, the general problem concerning how a questioner identifies his information-seeking goals and then acts to achieve them has so far remained unaddressed. Yet I suspect that we will ultimately require a better model of this process in order to understand the kinds of inference and reasoning abilities that cooperative respondents must possess.

We can at least partially describe the events leading up to Q's utterances using a conventional, commonsense model of Q as a goal-driven planning agent. It is reasonable to conjecture that the behavior of agents in the world is at least partially explained by their pursuit of rational goals. That is, agents identify desirable states of the world as their goals, and then proceed to form and execute plans that they believe will bring about those goals.[1] At least one of the reasons why agents interact with one another is to obtain information they perceive as needed to carry out their plans. For example, when an agent discovers that he is blocked from further progress in a plan, he will adopt a sub-goal to

[1] I am glossing over many details here because they do not contribute to the discussion. One such detail concerns the role of *intentionality* in rational planning behavior. For an excellent, rigorous analysis of the interrelationship of plans, goals, and intentions, the reader is referred to [Cohen 90a].

eliminate the obstacle, then form and execute a plan to that end. As Allen has observed [Allen 83], the lack of a particular piece of information can be one kind of impediment to an agent's progress. Q's goal in the present example is to speak to Kathy; his plan involves calling the hospital, since he seems to believe that Kathy is there (R can only determine this through the process of plan recognition). But Q has found himself blocked because he does not know the telephone number of the hospital where Kathy is staying.

Q might be able to obtain the telephone number in various ways, e.g., by looking in the telephone directory. But for whatever reason (convenience perhaps, or the absence of a telephone book), Q decides in this case to try to obtain the information from R. At this point Q must find a way to cause R to give him the desired information. In the example, Q has chosen to use linguistic acts to that end. Of course, this still does not tell us exactly how Q plans his two-utterance request given that he has recognized his need to know the hospital phone number. This is an open problem worthy of study.

The question of interest here concerns Q's purpose in uttering (72)a and the intended relationship of that utterance to (72)b. It seems that Q is stating the top-level goal of the plan he is pursuing. Assuming that Q's utterance was intended to cause R to believe that Q intends to talk to Kathy, what reason did Q have for trying to communicate that intention? Why did he not simply ask the question in (72)b and leave it at that?

There are various explanations one might consider. For example, Q might be trying to prevent R from jumping to any unwarranted conclusions. He might have reasoned that if he were to just ask the question and provide no motivation for it, she might become worried, supposing Q to be in need of hospitalization. Alternatively, what appears to be a goal statement might actually be the product of some very clever communicative planning. Notice that Q uses the definite noun phrase "the hospital." Assuming that Q and R share knowledge that Kathy was recently admitted to some particular hospital,[2] Q, in mentioning Kathy, could be trying by association to make that hospital salient in the conversation, thereby providing a convenient referent for the noun phrase. This is a particularly handy linguistic device, and would be especially useful if Q did not know the hospital's name or could not come up with a reasonably concise referring expression.

Other accounts are undoubtedly possible. However, to the extent that this issue has been considered at all, it has generally been assumed that Q chooses his utterances at least partly intending to make his underlying plan (the plan in which he is currently blocked) recognizable to R. Corresponding to this assumption, the process by which R infers Q's plan has been called *intended recognition*: recognition under the condition that Q is actively trying to make his plan recognizable to R [Cohen 81].[3]

The theory of intended recognition raises questions concerning both *why* and *how* questioners circumscribe the portions of their plans that they want respondents to know about. These are important questions worthy of careful study, and I believe that better models of the reasoning processes of questioners will be helpful in understanding the obligations and responsibilities of a cooperative respondent. That is, a better understanding of how questioners plan requests to support their goals in the world will surely help us develop better theories of cooperative responses.

[2] This was in fact the case in the real-life situation that gave rise to the dialogue.

[3] This is as opposed to *keyhole recognition*, in which the inferring agent deduces an actor's plans by observing the actor's actions as if through a keyhole. The actor is unaware that he is being observed, and hence makes no effort to assist the recognition process.

### 6.3.2 Mutual beliefs and cooperative responses

Theories of cooperative behavior will ultimately have to recognize the subtle role that mutual belief plays in the formulation of cooperative responses.[4] Consider the following scenario: A is attending a dinner party at B's home, and the two of them are together in B's kitchen (both A and B are female in this example). The other dinner guests, A's husband among them, are out in the dining room. It is dessert time and B is busy slicing up a cake and putting the pieces on plates. The following exchange ensues:[5]

(74) A: What's in the cake?
B: Don't worry—he can eat it.

Of critical importance to an understanding of this exchange is B's knowledge that A's husband, for religious reasons, will not eat foods containing certain proscribed ingredients. One question is whether A actually intended (i.e., whether she was making an effort to cause) B to recognize her higher purpose of determining whether the cake conformed to the requirements of her husband's religion. Why, for example, did she not simply ask, "Can my husband eat it?" This choice of query seems to at least partly depend upon whether A believed it was mutually believed that B knew that A's husband observed those dietary restrictions. The felicity of B's response would seem to depend upon her belief that it was mutually believed that she knew what those restrictions were and was able to evaluate for herself whether A's husband could in fact eat the cake. A complete account of the beliefs and inferences on both sides of this exchange would be most interesting indeed.

### 6.3.3 Correcting misconceptions

We saw in Chapter 2 that several researchers have studied misconception-correction techniques. Their efforts were concerned either with developing methods for *detecting* misconceptions (e.g., Kaplan [Kaplan 82], Mays [Mays 80]), or with characterizing corrective response strategies (e.g., McCoy [McCoy 85], Quilici [Quilici 88]).

A more principled, respondent-based approach would be to investigate the planned nature of misconception correction. Begin by assuming that the respondent has ascertained that the questioner incorrectly believes some proposition *P*. What constitutes a "correction" of that false belief? What knowledge and reasoning procedures are used by the respondent to compute a set of utterance acts that achieve the goal of correction? Several kinds of knowledge have been identified so far:

- beliefs about the "type" of the misconception (e.g., relating to some domain concept, or connected in some way to an intended plan);
- beliefs about entities that are salient in the discourse;
- beliefs about the (possibly erroneous) justifications that Q might have for believing *P*;

[4] Joshi has investigated the role of mutual belief in formulating answers to requests for descriptions of the referent of a referring expression [Joshi 82].

[5] This is a naturally-occurring exchange reported to me by Robert Rubinoff.

- beliefs about some related true proposition $P'$ whose truth implies the falsehood of $P$;
- beliefs derived as a side effect of identifying $P$ as a misconception.

In short, the research problem is to develop a model of the planning process that shows how R reasons about the discourse context, her private beliefs about the static and dynamic properties of the domain, her beliefs about Q's model of the world, and her beliefs about beliefs shared in common with Q, to select a set of linguistic acts that together function to correct Q's misconception.

### 6.3.4 Distinguishing cooperative reasoning from responding

In order to develop a better understanding of cooperative behavior, it may prove useful to try to separate two dimensions of a respondent's goal-driven behavior. The naturally-occurring examples considered in this dissertation provide evidence that respondents adopt goals to carry out various reasoning activities in support of their response goals; this behavior might be usefully distinguished from the processes of conveying some or all of the information so acquired to the questioner.

For example, in the course of planning a response, a respondent might decide to try to compute an alternative plan for a given goal, in order to compare it with the plan she has ascribed to the questioner. Since reasoning agents act by adopting and pursuing goals, that decision would be implemented by having the respondent adopt the goal of synthesizing a new plan. If that goal is achieved, the respondent might next adopt a goal of comparing the newly-computed plan with the questioner's plan. Ultimately, the respondent would have to decide what information to actually communicate to the questioner. The point is that we might want to separate the principles of cooperation that drive respondents to *reason* in the various ways they do from the principles that guide their choice of response content.

### 6.3.5 Alternative architectures

It may be worthwhile to consider alternatives to the model of high-level goal adoption I have proposed in this dissertation. According to my model, high-level response goals are adopted using a two-step process: first, goals that are well motivated on the basis of specific evidence found in the conversation model are proposed, becoming the Response Goal Proposer's *potential response goals.* Then other general reasoning procedures—which I have termed *screens of admissibility*—are brought to bear on that set of goals to select a subset to actually adopt. In proposing the use of screens of admissibility, I am attempting to accommodate both data and research results that suggest that respondents may, on the basis of other beliefs derived from or pertaining to the interaction, find general reasons for disregarding otherwise well-motivated response goals. For one example, recognition that the questioner's plan is ill formed may justify disregarding goals that have to do with providing information Q needs to instantiate one of its constituent actions. For another, a respondent may choose not to answer an explicit question if she can determine that providing that answer would conflict with one of her higher-priority goals (e.g., to maintain system security).

However, different goal-adoption strategies might be equally plausible. For example, there might be a way to order potential response goals according to some measure of importance. Then, rather than explicitly eliminating particular goals from consideration, the system could instead simply adopt the highest-ranked goal and attempt to satisfy it. Depending upon the results of that process, the remaining goals could be re-ranked and the new top-priority goal selected. This process could be repeated until some resource allotment was exceeded. While such a model could certainly be used, its consequences for response generation are unclear. In particular, it is not clear whether such an approach would provide a principled account for the omission of direct answers in the kinds of situations examined by Pollack and others. Nevertheless, the point is that until procedures are developed for evaluating CRG system architectures, it might be useful to explore the consequences of alternative designs to the one I have proposed.

## 6.4 Closing Remarks

Although I have focused my attention in this dissertation on the very general problem of producing cooperative responses to natural-language requests, the fundamental question of what actually makes a response "cooperative" remains murky. The many examples that were considered herein have demonstrated the profound complexity of cooperative dialogue. If this work has succeeded only in sharpening some of the research questions, a small but useful step forward has been achieved.

**Acknowledgments** The research work reported in this document has benefited greatly from interactions with many people during my years at Penn. First and foremost, I am indebted to my advisor Bonnie Webber for her guidance and unwavering support and encouragement throughout the development and writing of this thesis. Robert Rubinoff carefully and thoughtfully read countless drafts of the dissertation, and was a ever-ready source of insight and valuable feedback. Ellen Hays's editorial advice helped improve the readability of this document. Aravind Joshi's financial support and guidance at critical moments are gratefully acknowledged. Comments from the other members of my thesis committee, James Allen and Mitch Marcus, have helped clarify my thinking about issues being addressed in this work.

# Bibliography


[Allen 78] Allen, James F. and Perrault, C. Raymond. Participating in Dialogues: Understanding Via Plan Deduction. In *Proceedings of the 1978 Conference of the Canadian Society for Computational Studies of Intelligence.* 1978.

[Allen 80] Allen, James F. and Perrault, C. Raymond. Analyzing Intention in Utterances. *Artificial Intelligence* 15:143–178, 1980.

[Allen 83] Allen, James F. Recognizing Intentions from Natural Language Utterances. In Brady, M. and Berwick, R. C. (editors), *Computational Models of Discourse*, pages 107–166. The MIT Press, Cambridge, MA, 1983.

[Allen 89] Allen, James F. and Hinkelman, Elizabeth A. Using Structural Constraints for Speech Act Interpretation. In *Proceedings of the 1989 DARPA Workshop on Speech and Natural Language Processing.* October 1989.

[Allport 89] Allport, David and Kidd, Alison. *Using knowledge about search spaces to give co-operative responses.* Technical Report HPL-ISC-TM-89-129, Hewlett-Packard Laboratories, Bristol, England, August 1989.

[Appelt 85a] Appelt, Douglas E. Planning English Referring Expressions. *Artificial Intelligence* 26:1–33, 1985.

[Appelt 85b] Appelt, Douglas E. *Planning English Sentences.* Cambridge University Press, Cambridge, 1985.

[Austin 62] Austin, J. L. *How to Do Things with Words.* Oxford University Press, New York, 1962.

[Calistri 90] Calistri, Randall J. *Classifying and Detecting Plan-Based Misconceptions for Robust Plan Recognition.* PhD thesis, Department of Computer Science, Brown University, May 1990.

[Carberry 83] Carberry, Sandra. Tracking User Goals in an Information Seeking Environment. In *Proceedings of the* $3^{rd}$ *National Conference on Artificial Intelligence*, pages 59–63. 1983.

[Carberry 88] Carberry, Sandra. Plan Recognition and Its Use in Understanding Dialogue. In Kobsa, A. and Wahlster, W. (editors), *User Models in Dialog Systems.* Springer-Verlag, Berlin–New York, 1988.

[Carberry 90] Carberry, Sandra. Incorporating Default Inferences into Plan Recognition. In *Proceedings of the $9^{th}$ National Conference on Artificial Intelligence*, pages 471–478. 1990.

[Chapman 87] Chapman, David. Planning for Conjunctive Goals. *Artificial Intelligence* 32:333–377, 1987.

[Cheikes 87] Cheikes, Brant A. *An Architecture for Multi-Functional Natural Language Systems.* Technical Report MS-CIS-87-60, Department of Computer and Information Science, University of Pennsylvania, June 1987.

[Cheikes 89] Cheikes, Brant A. *The Architecture of a Cooperative Respondent.* Technical Report MS-CIS-89-13, Department of Computer and Information Science, University of Pennsylvania, February 1989.

[Chin 88a] Chin, David N. *Intelligent Agents as a Basis for Natural Language Interfaces.* PhD thesis, Computer Science Division, University of California, Berkeley, 1988.

[Chin 88b] Chin, David N. The UNIX Consultant as an Intelligent Agent. In *Proceedings of the Workshop on Architectures for Intelligent Interfaces: Elements and Prototypes*, pages 237–260. March 1988.

[Clifford 88] Clifford, James. Natural Language Querying of Historical Databases. *Computational Linguistics* 14(4):10–34, 1988.

[Cohen 78] Cohen, Philip R. *On Knowing What to Say: Planning Speech Acts.* Technical Report 118, Department of Computer Science, University of Toronto, January 1978.

[Cohen 79] Cohen, Philip R. and Perrault, C. Raymond. Elements of a Plan-Based Theory of Speech Acts. *Cognitive Science* 3(3):177–212, 1979.

[Cohen 81] Cohen, Philip R., Perrault, C. Raymond, and Allen, James F. Beyond Question-Answering. In Lehnert, W. and Ringle, M. (editors), *Strategies for Natural Language Processing.* Lawrence Erlbaum, Englewood Cliffs, 1981.

[Cohen 90a] Cohen, Philip R. and Levesque, Hector J. Persistence, Intention, and Commitment. In Cohen, P. R., Morgan, J., and Pollack, M. E. (editors), *Intentions in Communication*, pages 33–70. The MIT Press, Cambridge, MA, 1990.

[Cohen 90b] Cohen, Philip R. and Levesque, Hector J. Rational Interaction as the Basis for Communication. In Cohen, P. R., Morgan, J., and Pollack, M. E. (editors), *Intentions in Communication*, pages 221–255. The MIT Press, Cambridge, MA, 1990.

[Corella 85] Corella, Francisco. Semantic Retrieval and Levels of Abstraction. In Kerschberg, L. (editor), *Expert Database Systems.* Benjamin/Cummings, Menlo Park, CA, 1985.

[Eller 90] Eller, Rhonda and Carberry, Sandra. A meta-rule approach to dynamic plan recognition. In *Proceedings of the* $2^{nd}$ *International Workshop on User Modeling.* March 1990.

[Eugenio 87] Eugenio, Barbara Di. Cooperative Behavior in the FIDO System. *Information Systems* 12(3):295–316, 1987.

[Fikes 71] Fikes, Richard E. and Nilsson, Nils J. STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving. *Artificial Intelligence* 2:189–208, 1971.

[Finin 86] Finin, Tim, Joshi, Aravind K., and Webber, Bonnie L. Natural Language Interactions with Artificial Experts. *Proceedings of the IEEE* 74(7):921–938, 1986.

[Gal 85] Gal, Annie. A Natural Language Database Interface that Provides Cooperative Answers. In *Proceedings of the* $2^{nd}$ *Conference on Artificial Intelligence Applications*, pages 352–357. 1985.

[Gal 88] Gal, Annie. *Cooperative Responses in Deductive Databases.* PhD thesis, Department of Computer Science, University of Maryland, College Park, July 1988.

[Goldman 70] Goldman, Alvin I. *A Theory of Human Action.* Prentice-Hall, Englewood Cliffs, 1970.

[Green 63] Green, Jr., Bert F., Wolf, Alice K., Chomsky, Carol, and Laughery, Kenneth. BASEBALL: An Automatic Question Answerer. In Feigenbaum, E. and Feldman, J. (editors), *Computers and Thought.* McGraw-Hill, New York, 1963.

[Grice 75] Grice, H. Paul. Logic and Conversation. In *Syntax and Semantics 3: Speech Acts.* Academic Press, New York, 1975.

[Grosz 77] Grosz, Barbara J. *The Representation and Use of Focus in Dialogue Understanding.* PhD thesis, Computer Science Division, University of California, Berkeley, June 1977.

[Grosz 81] Grosz, Barbara J. Focusing and description in natural language dialogues. In Joshi, A. K., Webber, B. L., and Sag, I. (editors), *Elements of Discourse Understanding*, pages 84–105. Cambridge University Press, 1981.

[Grosz 86] Grosz, Barbara J. and Sidner, Candace L. Attention, Intentions, and the Structure of Discourse. *Computational Linguistics* 12(3):175–204, 1986.

[Grosz 87] Grosz, Barbara J., Appelt, Douglas E., Martin, Paul A., and Pereira, Fernando C. N. TEAM: An Experiment in the Design of Transportable Natural-Language Interfaces. *Artificial Intelligence* 32:173–243, 1987.

[Harris 73] Harris, L. R. *The ROBOT System: Natural Language Processing Applied to Database Query.* Technical Report TR78-1, Department of Mathematics, Dartmouth College, Hanover, NH, 1973.

[Hendrix 78] Hendrix, Gary G., Sacerdoti, Earl D., Sagalowicz, Daniel, and Slocum, Jonathan. Developing a Natural Language Interface to Complex Data. *ACM Transactions on Database Systems* 3(3):105–147, 1978.

[Hinkelman 89] Hinkelman, Elizabeth A. and Allen, James F. Two Constraints on Speech Act Ambiguity. In *Proceedings of the* $27^{th}$ *Annual Meeting of the Association for Computational Linguistics*, pages 212–219. 1989.

[Hinkelman 90] Hinkelman, Elizabeth A. *Linguistic and Pragmatic Constraints on Utterance Interpretation.* PhD thesis, Department of Computer Science, University of Rochester, May 1990.

[Hintikka 62] Hintikka, Jaakko. *Knowledge and Belief.* Cornell University Press, Ithaca, 1962.

[Hintikka 69] Hintikka, Jaakko. Semantics for propositional attitudes. In *et al.*, J. W. Davis (editor), *Philosophical logic.* D. Reidel Publishing Company, Dordrecht, 1969.

[Hovy 88] Hovy, Eduard H. *Generating Natural Language Under Pragmatic Constraints.* Lawrence Erlbaum, Hillsdale, NJ, 1988.

[Joshi 82] Joshi, Aravind K. Mutual Beliefs in Question Answering Systems. In Smith, N. (editor), *Mutual Belief.* Academic Press, New York, 1982.

[Joshi 84a] Joshi, Aravind K., Webber, Bonnie L., and Weischedel, Ralph. Living Up to Expectations: Computing Expert Responses. In *Proceedings of the* $4^{th}$ *National Conference on Artificial Intelligence.* 1984.

[Joshi 84b] Joshi, Aravind K., Webber, Bonnie L., and Weischedel, Ralph. Preventing False Inferences. In *Proceedings of the* $10^{th}$ *International Conference on Computational Linguistics*, pages 134–138. 1984.

[Joshi 87] Joshi, Aravind K., Webber, Bonnie L., and Weischedel, Ralph. Some Aspects of Default Reasoning in Interactive Discourse. In Reilly, R. G. (editor), *Communication Failure in Dialogue and Discourse*, pages 213–219. Elsevier, 1987.

[Kaplan 81] Kaplan, S. Jerrold. Appropriate Responses to Inappropriate Questions. In Joshi, A. K., Webber, B. L., and Sag, I. (editors), *Elements of Discourse Understanding*, Chapter 5, pages 127–144. Cambridge University Press, 1981.

[Kaplan 82] Kaplan, S. Jerrold. Cooperative Responses from a Portable Natural Language Database Query System. In Brady, M. (editor), *Computational Models of Discourse.* The MIT Press, Cambridge, MA, 1982.

[Kass 88] Kass, Robert and Finin, Tim. Modeling the User in Natural Language Systems. *Computational Linguistics* 14(3):5–22, 1988.

[Kautz 87] Kautz, Henry A. *A Formal Theory of Plan Recognition.* PhD thesis, Department of Computer Science, University of Rochester, May 1987.

[Kidd 90] Kidd, Alison L. Deriving Principles for Generating Co-operative Responses: An Experiment. In *Proceedings of the 1990 Human Factors in Computer Systems Conference.* 1990.

[Konolige 89] Konolige, Kurt and Pollack, Martha E. Ascribing Plans to Agents: Preliminary Report. In *Proceedings of the* $11^{th}$ *International Conference on Artificial Intelligence*, pages 924–930. 1989.

[Lewis 82] Lewis, B. and Goldberg, E. EMACS Transcripts. 1982. Collected in Spring 1982 at the University of Pennsylvania.

[Litman 84a] Litman, Diane J. and Allen, James F. A Plan Recognition Model for Clarification Subdialogues. In *Proceedings of the* $10^{th}$ *International Conference on Computational Linguistics*, pages 302–311. 1984.

[Litman 84b] Litman, Diane J. and Allen, James F. *A Plan Recognition Model for Subdialogues in Conversations.* Technical Report 141, Department of Computer Science, University of Rochester, 1984.

[Luria 88] Luria, Marc. *Knowledge Intensive Planning.* PhD thesis, Computer Science Division, University of California, Berkeley, July 1988.

[Malhotra 75] Malhotra, A. *Design criteria for a knowledge-based English language system for management: an experimental analysis.* Technical Report MAC TR-146, Massachusetts Institute of Technology, 1975.

[Mays 80] Mays, Eric. Failures in natural language systems: application to data base query systems. In *Proceedings of the* $1^{st}$ *National Conference on Artificial Intelligence.* 1980.

[Mays 82] Mays, Eric, Joshi, Aravind K., and Webber, Bonnie L. Taking the Initiative in Natural Language Data Base Interactions: Monitoring as Response. In *Proceedings of the* $5^{th}$ *European Conference on Artificial Intelligence.* 1982.

[Mays 84a] Mays, Eric. *A Modal Temporal Logic for Reasoning about Changing Databases with Applications to Natural Language Question Answering.* PhD thesis, Department of Computer and Information Science, University of Pennsylvania, 1984.

[Mays 84b] Mays, Eric. A Temporal Logic for Reasoning about Changing Data Bases in the Context of Natural Language Question-Answering. In *Proceedings of the* $1^{st}$ *International Workshop on Expert Database Systems*, pages 238–257. Kiawah Island, SC, October 1984.

[McCoy 85] McCoy, Kathleen F. *Correcting Object-Related Misconceptions.* PhD thesis, Department of Computer and Information Science, University of Pennsylvania, December 1985.

[McCoy 88] McCoy, Kathleen F. Reasoning on a Highlighted User Model to Respond to Misconceptions. *Computational Linguistics* 14(3):52–63, 1988.

[McKeown 79] McKeown, Kathleen R. Paraphrasing Using Given and New Information in a Question-Answer System. In *Proceedings of the* $17^{th}$ *Annual Meeting of the Association for Computational Linguistics*, pages 67–72. 1979.

[McKeown 85] McKeown, Kathleen R. *Text Generation: Using Discourse Strategies and Focus Constraints to Generate Natural Language Text.* Cambridge University Press, Cambridge, 1985.

[Mercer 84] Mercer, R. E. and Rosenberg, R. S. Generating Corrective Answers by Computing Presuppositions of Answers, Not of Questions. In *Proceedings of the 1984 Conference of the Canadian Society for Computational Studies of Intelligence*, pages 16–19. 1984.

[Moore 89] Moore, Johanna D. and Paris, Cécile L. Planning Text for Advisory Dialogues. In *Proceedings of the* $27^{th}$ *Annual Meeting of the Association for Computational Linguistics*, pages 203–211. 1989.

[Motro 84] Motro, Amihai. Query Generalization: A Technique for Handling Query Failure. In *Proceedings of the* $1^{st}$ *International Workshop on Expert Database Systems*, pages 314–325. Kiawah Island, SC, 1984.

[Nii 86a] Nii, H. Penny. Blackboard Systems: Blackboard Application Systems, Blackboard Systems from a Knowledge Engineering Perspective. *AI Magazine* 7(3):82–106, 1986.

[Nii 86b] Nii, H. Penny. Blackboard Systems: The Blackboard Model of Problem Solving and the Evolution of Blackboard Architectures. *AI Magazine* 7(2):38–53, 1986.

[Pollack 86] Pollack, Martha E. *Inferring Domain Plans in Question-Answering.* PhD thesis, Department of Computer and Information Science, University of Pennsylvania, May 1986.

[Quilici 88] Quilici, Alex, Dyer, Michael, and Flowers, Margot. Detecting and Responding to Plan-Oriented Misconceptions. In Kobsa, A. and Wahlster, W. (editors), *User Models in Dialog Systems.* Springer-Verlag, Berlin–New York, 1988.

[Ramshaw 89] Ramshaw, Lance A. *Pragmatic Knowledge for Resolving Ill-formedness.* PhD thesis, Department of Computer and Information Science, University of Delaware, May 1989.

[Ramshaw 91] Ramshaw, Lance A. A three-level model for plan exploration. In *Proceedings of the $29^{th}$ Annual Meeting of the Association for Computational Linguistics*, pages 39–46. 1991.

[Reiter 80] Reiter, Ray. A logic for default reasoning. *Artificial Intelligence* 13:81–132, 1980.

[Rich 79] Rich, Elaine. *Building and Exploiting User Models.* PhD thesis, Department of Computer Science, Carnegie-Mellon University, 1979.

[Rissland 84] Rissland, E., Valcarce, E., and Ashley, K. Explaining and Arguing with Examples. In *Proceedings of the $4^{th}$ National Conference on Artificial Intelligence*, pages 288–294. 1984.

[Rubinoff 90] Rubinoff, Robert. *Natural Language Generation as an Intelligent Activity.* Technical Report MS-CIS-90-32, Department of Computer and Information Science, University of Pennsylvania, May 1990.

[Searle 69] Searle, John R. *Speech acts: An essay in the philosophy of language.* Cambridge University Press, Cambridge, 1969.

[Searle 71] Searle, John R. What is a Speech Act? In Searle, J. R. (editor), *The Philosophy of Language*, pages 23–38. Oxford University Press, London, 1971.

[Searle 75] Searle, John R. Indirect Speech Acts. In *Syntax and Semantics 3: Speech Acts.* Academic Press, New York, 1975.

[Sidner 81] Sidner, Candace L. and Israel, David. Recognizing Intended Meaning and Speaker's Plans. In *Proceedings of the $7^{th}$ International Conference on Artificial Intelligence*, pages 203–208. 1981.

[Sidner 83] Sidner, Candace L. What the Speaker Means: The Recognition of Speakers' Plans in Discourse. In Cercone, N. J. (editor), *Computational Linguistics.* Pergamon Press, Oxford, 1983.

[Stenton 86] Stenton, S. P. Dialogue management for co-operative knowledge based systems. *Knowledge Engineering Review* 2(2):99–122, 1986.

[vanBeek 87] van Beek, Peter. A model for generating better explanations. In *Proceedings of the $25^{th}$ Annual Meeting of the Association for Computational Linguistics*, pages 215–220. 1987.

[vanBeek 90] van Beek, Peter and Cohen, Robin. Resolving Plan Ambiguity for Response Generation. In *Proceedings of the $5^{th}$ International Workshop on Natural Language Generation*, pages 144–149. June 1990.

[Wahlster 83] Wahlster, Wolfgang, Marburger, Heinz, Jameson, Anthony, and Busemann, Stephan. Over-Answering Yes-No Questions: Extended responses in a NL interface to a vision system. In *Proceedings of the $8^{th}$ International Conference on Artificial Intelligence*, pages 643–646. 1983.

[Waltz 78] Waltz, D. L. An English Language Question Answering System for a Large Relational Data Base. *Communications of the ACM* 21(7):526–539, 1978.

[Webber 84] Webber, Bonnie L. and Finin, Tim. In Response: Next Steps in Natural Language Interaction. In Reitman, W. (editor), *Artificial Intelligence Applications for Business*, pages 211–234. Ablex, Norwood, NJ, 1984.

[Webber 86] Webber, Bonnie L. Questions, Answers, and Responses: A Guide for Knowledge Based Systems. In Brodie, M. and Mylopoulos, J. (editors), *On Knowledge Base Systems.* Springer-Verlag, Amsterdam, 1986.

[Wilensky 87] Wilensky, Robert. *Some Problems and Proposals for Knowledge Representation.* Technical Report UCB/CSD 87/351, Computer Science Division, University of California, Berkeley, 1987.

[Wilensky 88] Wilensky, Robert, Chin, David N., Luria, Marc, Martin, James, Mayfield, James, and Wu, Dekai. The Berkeley UNIX Consultant Project. *Computational Linguistics* 14(4):35–84, 1988.

[Woods 72] Woods, William A., Kaplan, R. M., and Nash-Webber, Bonnie L. *The Lunar Sciences Natural Language Information System: Final Report.* Technical Report 2378, Bolt, Beranek and Newman, Cambridge, MA, 1972.

[Woods 77] Woods, William A. *Semantics and Quantification in Natural Language Question Answering.* Technical Report 3687, Bolt, Beranek and Newman, Cambridge, MA, 1977.