Spoken Natural Language Dialog Systems

A PRACTICAL APPROACH

Ronnie W. Smith and
D. Richard Hipp

Spoken Natural Language Dialog Systems

This page intentionally left blank

Spoken Natural Language Dialog Systems: A Practical Approach

RONNIE W. SMITH
East Carolina University

D. RICHARD HIPP
Hipp, Wyrick and Company, Inc.

New York Oxford
OXFORD UNIVERSITY PRESS
1994

Oxford University Press

Oxford New York
Athens Auckland Bangkok Bombay
Calcutta Cape Town Dar es Salaam Delhi
Florence Hong Kong Istanbul Karachi
Kuala Lumpur Madras Madrid Melbourne
Mexico City Nairobi Paris Singapore
Taipei Tokyo Toronto

and associated companies in
Berlin Ibadan

Published by Oxford University Press, Inc.,
200 Madison Avenue, New York, New York 10016

Library of Congress Cataloging-in-Publication Data
Smith, Ronnie W.
Spoken natural language dialog systems: a practical approach /
Ronnie W. Smith, D. Richard Hipp.
p. cm. Includes bibliographical references and index.
ISBN 0-19-509187-6
1. Speech processing systems.
2. Natural language processing (Computer science).
I. Hipp, D. Richard. II. Title.
TK7895.S65S62 1994 006.4'54—dc20 94-17053

9 8 7 6 5 4 3 2 1

Printed in the United States of America
on acid-free paper

Acknowledgments

This book is based on research conducted primarily while the authors were graduate students at Duke University. Our deepest thanks goes to our dissertation advisor and good friend, Dr. Alan Biermann, who provided the vision, encouragement, and constructive questions and suggestions that made this research and book possible. Another project collaborator, Dr. Robert Rodman of North Carolina State University, provided many helpful comments about the directions this research should undertake. The scope of the experimental results obtained would not have been possible without the assistance of Dr. Ruth Day of Duke University who provided us with innumerable suggestions and comments concerning the experimental design.

We are also thankful to several fellow students for help. Most notably, Dania Egedi implemented a key portion of the system for resolving domain ambiguities and assisted in carrying out the experiment. Robin Gambill implemented the computer's utterance generator. We also had productive discussions about this research with Barry Koster, Al Nigrin, and Curry Guinn. We are also thankful to the volunteers who helped us test system prototypes at various times. These include Michael Brown, Dave Reed, Dania Egedi, Boyce Byerly, Curry Guinn, Wendy Heine, and Anna Harwell.

Development of the final manuscript has benefited greatly from the support and suggestions of our editor Don Jackson and from the tireless efforts of our copy-editor, Virginia Holman. Special thanks go to the Department of Computer Science at Duke University for its continued support of our work and to the Department of Mathematics at East Carolina University. We would also like to thank Dr. Steve Gordon, Curry Guinn, and Dr. Alan Biermann, who have provided useful suggestions about the book's content. Our appreciation also goes to the anonymous reviewers whose comments assisted us greatly in our final revisions. Any remaining errors are, of course, the sole responsibility of the authors.

Financial support for this work has come from National Science Foundation grants IRI-88-03802 and IRI-92-21842, Dr. Charles Putman of Duke University, and the Research and Creative Activity Committee of East Carolina University. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

This page intentionally left blank

Contents

Spoken Natural Language Dialog Systems

This page intentionally left blank

Chapter 1

Achieving Spoken Communication with Computers

The most sophisticated and efficient means of communication between humans is spoken *natural language* (NL). It is a rare circumstance when two people choose to communicate via another means when spoken natural language is possible. Ochsman and Chapanis [OC74] conducted a study involving two person teams solving various problems using restricted means of communication such as typewriting and video, typewriting only, handwriting and video, voice and video, voice only, etc. Their conclusion included the following statement.

> The single most important decision in the design of a telecommunications link should center around the inclusion of a voice channel. In the solution of factual real-world problems, little else seems to make a demonstrable difference.

Thus, it would seem desirable to develop computer systems that can also communicate with humans via spoken natural language dialog. Furthermore, recent reports from the research community in speech recognition [Adv93] indicate that accuracy levels in speaker-independent continuous speech recognition have reached a threshold where practical applications of spoken natural language are viable.

This book addresses the dialog issues that must be resolved in building effective spoken natural language dialog systems—systems where both the human and computer interact via spoken natural language.[1] We present an architecture for dialog processing for which an implementation in the equipment repair domain has been constructed that exhibits a number of behaviors required for efficient human-machine dialog. These behaviors include the following.

- Problem solving to achieve a target goal.
- The ability to carry out subdialogs to achieve appropriate subgoals and to pass control arbitrarily from one subdialog to another.
- The use of a user model to enable useful verbal exchanges and to inhibit unnecessary ones.
- The ability to use context dependent expectations to correct speech recognition and track user movement to new subdialogs.
- The ability to vary the task/dialog initiative from strongly computer controlled to strongly user controlled or somewhere in between.

[1] We do not preclude typewritten interaction, but we focus on phenomena of spoken interaction.

The ability to vary initiative is called *variable initiative* dialog, and represents a major advance in the utility of natural language systems. With such capability, a natural language system can effectively communicate with both task novices and experts as well as users with intermediate levels of expertise. Task novices require a computer-controlled dialog to lead them through the details. Task experts need occasional advice while pursuing their own task strategies. Users with intermediate expertise may be capable of carrying out portions of the task with little assistance, but they may also need detailed help during unfamiliar stages. A variable initiative dialog system is required for effective interaction in all these situations.

The following sample dialogs recorded on the videotape [HS91] illustrate the flexibility of a variable initiative dialog system. These dialogs were obtained from usage of the implemented system by an experimental subject that had no special knowledge about the implemented system or circuit repair. In the first dialog, the computer controls the dialog while in the second dialog, the system operates in such a way as to allow an experienced user to take the initiative and to control the dialog. The words in parenthesis beside or below a user utterance represent the actual sequence of words that the speech recognizer sent to the dialog system for analysis.

Dialog 1: Computer Controlled—elapsed time 4 minutes 35 seconds

computer: This is the circuit fix it shop. How may I help you?
user: I want to fix this circuit. (power a six a circuit)
Due to speech recognition errors, the system understood the meaning of this utterance as "There is a wire on connector six nine." This could not be understood in any active context.
computer: I do not understand. Please repeat what you said.
user: I want to fix this circuit. (i want a fix a circuit)
computer: What is the ID of the circuit?
user: RS111. (rs111)
computer: I am familiar with that circuit. The LED is supposed to be displaying alternately flashing one and seven.
user: The LED is displaying nothing. (be led a displaying nothing)
The computer must now supply context in order to get the user to make the desired observation.
computer: What is the switch at when the LED is off?
user: The switch is down. (the switch yes down)
computer: Put the switch up.
user: Done. (done)
computer: Put the knob to one zero.
user: Which knob? (which knob)
There are two knobs on the circuit board.
computer: Near the top boundary and in the center is a green region. Now, in the middle right of this green region is the control knob.
user: The knob is at one zero. (the knob and add one zero)
computer: What is the LED displaying?
user: The LED is displaying a one and seven at the same time.
(be led a displaying one an seven at the same time)

computer: Put the knob to zero.
user: Done. (done)
computer: What is the LED displaying?
user: The same thing. (the same same)

Due to a speech recognition error, the computer was uncertain about what the user said, and therefore verifies that it understood correctly. Notice how the machine fills in an elliptic portion of the user's meaning: "the LED is displaying."

computer: Did you mean to say the LED is displaying the same thing?
user: Yes. (yes)
computer: Is there a wire between connector six eight and connector eight seven?
user: Yes. (yes)
computer: Is there a wire between connector three zero and connector seven zero?
user: No. (know)

There is no (know?) phonetic difference between "know" and "no." Dialog context must be used to determine the correct interpretation.

computer: Add a wire between connector three zero and connector seven zero.
user: Done. (done)
computer: Put the knob to one zero.
user: Done. (done)
computer: What is the LED displaying?
user: The LED is displaying an alternately flashing one and seven.
(be led it displaying alternately flashing one and seven)
computer: Is the one on the LED displaying for a longer period of time?
user: No. (know)
computer: Is the seven on the LED displaying for a longer period of time?
user: No. (know)
computer: Put the knob to zero.
user: Done. (done)
computer: What is the LED displaying?
user: The one is flashing for a longer period of time.
(one is flashing forth longer in a time)

The word "period" is not in the vocabulary of the speech recognizer. As a result, the output of the speech recognizer was badly mangled. The parser was only able to make sense of the three words "one is flashing" and took its meaning from them.

computer: Did you mean to say the LED is flashing a one?
user: The LED is flashing a one and seven.
(be led it flashing one and seven)
computer: Is the one on the LED displaying for a longer period of time?
user: Yes. (yes)
computer: Put the switch down.
user: Done. (done)
computer: What is the LED displaying?
user: The LED is displaying nothing. (be led a displaying nothing)
computer: The circuit is working correctly. Good-bye.

In the second dialog the user now has the initiative. Notice the two places where the user makes an abrupt shift of context, which the computer tracks correctly.

Dialog 2: User Controlled—elapsed time 1 minute 28 seconds

computer: This is the circuit fix it shop. How may I help you?

user: I want to fix this rs111. (i want fix this rs111)

computer: I am familiar with that circuit. The LED is supposed to be displaying alternately flashing one and seven.

user: The LED is displaying an alternately flashing one and the top corner of a seven.
(be led it displaying alternately flashing one in top corner a seven)
With the user in control the computer does not ask about the switch position as in the previous dialog, but instead bases its response on the perceived user focus.

computer: There is supposed to be a wire between connector nine nine and connector one zero zero.

user: It is there. (it in there)

computer: There is supposed to be a wire between connector nine eight and connector one zero two.

user: There is no wire on connector one zero four.
(stays know wire i connector one zero for)
While checking for the wire between 98 and 102, the user notices that there is no wire on the nearby connector 104. She makes a change in context to alert the machine to this fact, which she now suspects is the cause of the problem.

computer: There is supposed to be a wire between connector one zero four and connector one zero two.

user: The circuit is working. (the circuit is working)
After finding that the wire between 102 and 104 is missing, the user is able to finish repairing the circuit on her own. She makes another context shift to notify the machine.

computer: Glad to have been of assistance. Good-bye.

Obtaining such robust and efficient behavior from a spoken natural language dialog system requires integrating methods for achieving each of the target behaviors previously mentioned into a single self-consistent mechanism. While the individual behaviors have been well studied, it is the development of the mechanism for integration that is the major research contribution presented in this book. In addition, we present results on system effectiveness, usability, and human performance based on a formal experiment involving usage of the system by eight different subjects in 141 dialogs.

1.1 Problem Solving Environment: Task-Oriented Dialogs

Task-oriented dialogs are dialogs about the performance of a task that occurs as the task is being performed. The associated problem solving environment studied in this research is characterized by the following.

- The user is a person with the ability to carry out all the required sensory and mechanical operations to solve the problem, but with insufficient knowledge to solve the problem without assistance.

- The computer has complete knowledge about the task and its purpose. This means that the computer has sufficient knowledge to perform the task if it can perform the required sensory and mechanical actions and that the assistance to be provided is based on knowing the purpose of the underlying task. Consequently, the role of the dialog is to ensure that the necessary data is obtained and the proper actions performed. This contrasts with isolated fact retrieval or database query systems whose cooperativeness can extend beyond simple question-answering only to consideration of presuppositions (see Kaplan [Kap82] for example).

- The computer will communicate with the user via natural language. The requirement for sufficient cooperativeness necessitates such an interface to allow for all the functions of human communication as described by Sowa [Sow84]. Among other things, the user may wish to select between a possible set of descriptions, express a relationship that provides a description, issue a command to the computer to perform some action, ask a question about some information, or explain the motivations for performing some action. To allow the user and computer to communicate in all these necessary ways requires the use of natural language.

In such an environment, the human and computer must cooperate to solve the problem. Furthermore, this cooperation requires the use of natural language dialog in order to succeed. For these dialogs Grosz [Gro78] notes that *the structure of a dialog mirrors the structure of the underlying task.* Since tasks normally follow a well-structured set of operations, dialogs about tasks should also be well-structured. It will be seen that exploitation of the close relationship between dialog and task structure is crucial for obtaining the efficiency in human-computer dialog that is ubiquitous in human-human dialog.

1.2 Integrating Dialog with Task Assistance: The Target Behaviors

The purpose of the architecture is to deliver to users in real time the previously listed behaviors that are needed for efficient human-machine dialog. The difficulty in developing the architecture is that the phenomena associated with task assistance are independent of the method of communication. Successful task assistance (i.e. domain problem solving) can be accomplished without any natural language interaction. However, as noted at the beginning of the chapter, natural language interaction is likely to be the most efficient form of communication between domain problem solver and human user. Furthermore, coherent natural language dialog requires consideration of the dialog's context. Consequently, a connection must be made between the task assistance context and the dialog processing architecture. As illustrated in figure 1.1 and to be shown in chapters 3 and 4, the Missing Axiom Theory for language use offers a connection between task assistance and dialog.

1.2.1 Problem Solving to Achieve a Goal

Efficient dialog requires that each participant understand the purpose of the interaction and be able to cooperate in its achievement. This is captured by the *intentional structure* of Grosz and Sidner [GS86], the description of the underlying goal-oriented purposes for engaging in the overall dialog, and the needed subdialogs. Required facilities include: (1) a domain problem solver that can suggest the necessary actions; and (2) a mechanism for determining when these actions are completed.

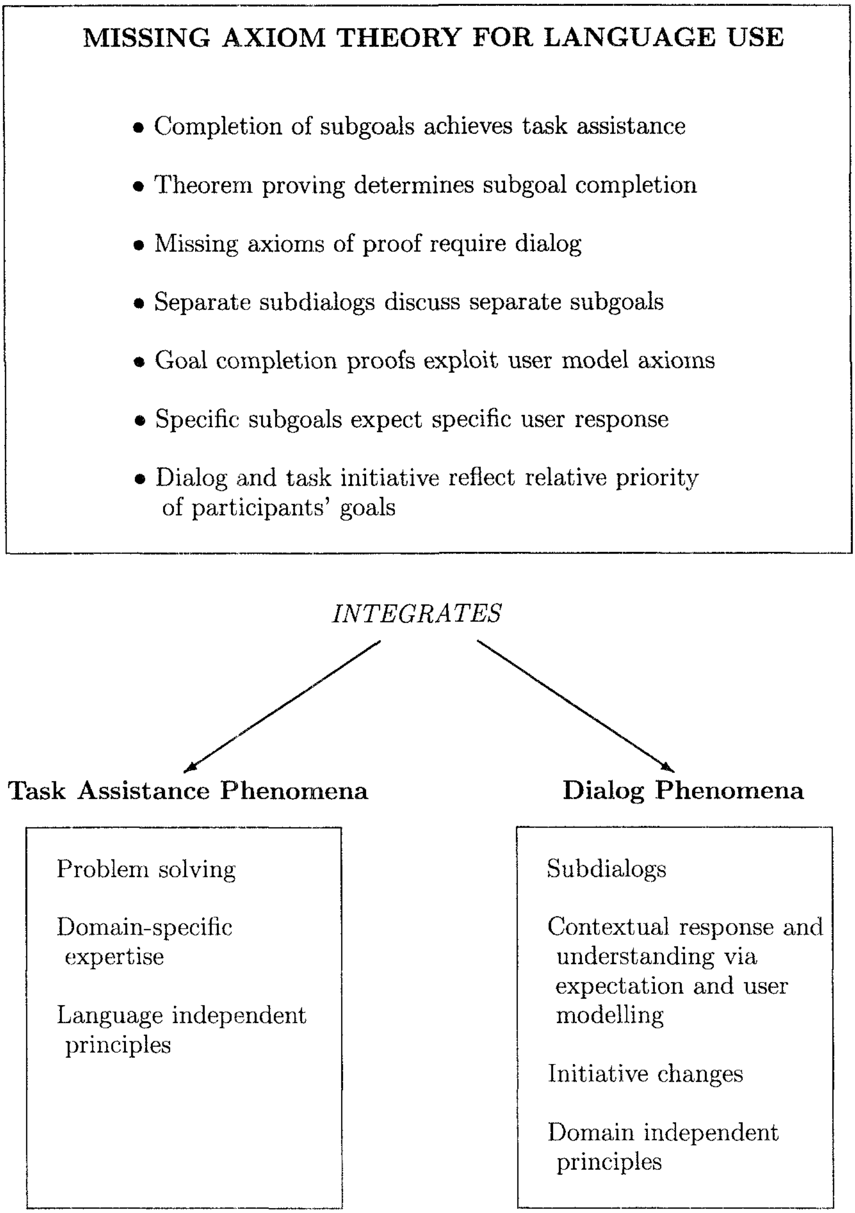
Developing a computational model that generalizes over task-oriented dialogs requires a general notion of task processing. The framework adopted here uses standard artificial intelligence (AI) planning terminology (see Nilsson [Nil80] for example) in saying that *actions* change the world state where the world state can be described by physical and/or mental state descriptions. During the course of a task, both the user and computer will have various *goals* to accomplish different actions or achieve different states. The task is accomplished by carrying out a sequence of actions that result in an appropriate world state as defined by the task.

Because variable initiative dialogs are allowed, the sequence of actions that occurs during the task may vary significantly during different executions of the task. This is especially true in repair tasks, where differences in the error source may require various diagnostic and corrective actions. When the computer has more initiative, the computer may require that specific actions be performed. Conversely, when the user has more initiative, the computer may merely offer recommendations or provide other relevant data.

Regardless of the lofty goals of a general abstract theory for task processing, a working system must contain domain-specific knowledge. How can a general theory be combined with a concrete implementation? As shown in chapter 3, a separation is required between the dialog processing component of the system and the domain specific task processing component. The general theory must provide a standardized method for communication between the two that facilitates successful task assistance.

1.2.2 Subdialogs and Effective Movement Between Them

An alternative title for this section could be, "Why Do We Say Anything?" Efficient human dialog is usually segmented into utterance sequences, *subdialogs*, that are aimed at achieving individual subgoals. These are called "segments" by Grosz and Sidner [GS86] and constitute the *linguistic structure* defined in their paper. The global goal is approached by a series of attempts at subgoals. Each attempt involves a set of interactions that constitutes a subdialog. We adopt the view that when an action is being attempted, the primary role of language is assistance in completing the action. To provide a well established computational framework, action completion will be defined by theorems, and determination of action completion will be accomplished by carrying out a proof of the appropriate theorem. With this viewpoint, the role of language



MISSING AXIOM THEORY FOR LANGUAGE USE

- Completion of subgoals achieves task assistance
- Theorem proving determines subgoal completion
- Missing axioms of proof require dialog
- Separate subdialogs discuss separate subgoals
- Goal completion proofs exploit user model axioms
- Specific subgoals expect specific user response
- Dialog and task initiative reflect relative priority of participants' goals

INTEGRATES

Task Assistance Phenomena

Problem solving

Domain-specific expertise

Language independent principles

Dialog Phenomena

Subdialogs

Contextual response and understanding via expectation and user modelling

Initiative changes

Domain independent principles

Figure 1.1
Integrated Processing Overview

is to supply missing axioms for completing the proof, and the utterances associated with a particular theorem constitute a subdialog. Consequently, the theorem prover must be able to suspend itself when it encounters a missing axiom to request its acquisition from an outside source (i.e. dialog). This view summarizes our *Missing Axiom Theory* for language use. As will be seen throughout the book, this theory is the key to achieving integrated dialog processing.

An aggressive strategy for successful task completion is to choose the subgoals judged most likely to lead to success and carry out their associated subdialogs. As the system proceeds on a given subdialog, it should always be ready to abruptly change to another subdialog if it suddenly seems more appropriate. This leads to the fragmented style that so commonly appears in efficient human communication. A subdialog is opened which leads to another, then another, then a jump to a previously opened subdialog, etc., in an unpredictable order until all necessary subgoals have been completed. Thus, the theorem prover must be capable of arbitrarily suspending the proof of one theorem to resume proving another as directed by the overall dialog processing mechanism. Furthermore, the theorem prover must be able to alter the structure of proofs that are being attempted to allow arbitrary clarification subdialogs.

1.2.3 Accounting for User Knowledge and Abilities

Cooperative problem solving involves maintaining a dynamic profile of user knowledge, termed a *user model*. The user model specifies information needed for efficient interaction with the conversational partner. Its purpose is to indicate what needs to be said to the user to enable the user to function effectively. It also indicates what should be omitted because of existing user knowledge.

Because considerable information is exchanged during the dialog, the user model changes continuously. Mentioned facts are stored in the model as known to the user and are not repeated. Previously unmentioned information may be assumed to be unknown and may be explained as needed. Questions from the user may indicate lack of knowledge and result in the removal of items from the user model.

To integrate the user model with the Missing Axiom Theory, the user model must also be represented as axioms. In this case, the axioms are about a particular user, what the user knows or believes, what the user can do, etc. Consequently, the user model can be utilized in a natural fashion in proving completion of actions. Where the user model indicates the user knowledge is adequate, no language interaction is needed. Where it is inadequate, the missing axiom indicates the need for dialog.

1.2.4 Expectation of User Input

Since all interactions occur in the context of a current subdialog, the user's input is far more predictable than would be indicated by a general grammar for English. In fact, the current subdialog specifies the *focus* of the interaction; the set of all objects and actions that are locally appropriate. This is the *attentional structure* described by Grosz and Sidner [GS86], and its most important function is to predict the content of user utterances. For example, if the user is asked to measure a voltage, the user's response may refer to the voltmeter, leads, voltage range, locations of measurement points, or the resulting measurement.

Therefore, based on the relevant missing axiom, the subdialog structure provides a set of expected utterances at each point in the dialog, and these have two important roles.

- The expected utterances provide strong guidance for the speech recognition system so that error correction can be enhanced. Where ambiguity arises, recognition can be biased in the direction of meaningful statements in the current context. In conjunction with the theory of parsing spoken natural language presented in chapter 5, this mechanism enabled the implemented system to understand correctly 81% of 2840 utterances spoken by experimental subjects although only 50% of the utterances were recognized correctly word for word.

- The expected utterances from subdialogs other than the current one can indicate that a shift from the current subdialog is occurring. Thus, expectations are one of the primary mechanisms needed for tracking the dialog when subdialog movement occurs. This is known elsewhere as the *plan recognition* problem, and it has received much attention in recent years.[2]

1.2.5 Variable Initiative

A real possibility in a cooperative interaction is that the user's problem solving ability, either on a given subgoal or on the global task, may exceed that of the machine. When this occurs, an efficient interaction requires that the machine yield control so that the more competent partner can lead the way to the fastest possible solution. Thus, the machine must not only be able to carry out its own problem solving process and direct the user toward task completion, but also be able to yield to the user's control and respond cooperatively as needed. This is a variable initiative dialog. As a pragmatic issue, we have found that at least four dialog initiative *modes* are useful.

[2] Carberry [Car90] describes recent work in plan recognition and also provides an extensive bibliography and review that includes a summary description of important work by Allen and Perrault [AP80] and Litman and Allen [LA87].

- *Directive*—The computer has complete dialog control. It recommends a subgoal for completion and will use whatever dialog is necessary to obtain the needed item of knowledge related to the subgoal.

- *Suggestive*—The computer still has dialog control, but not as strongly. The computer will suggest which subgoal to perform next, but it is also willing to change the direction of the dialog according to stated user preferences.

- *Declarative*—The user has dialog control, but the computer is free to mention relevant, though not required, facts as a response to the user's statements.

- *Passive*—The user has complete dialog control. The computer responds directly to user questions and passively acknowledges user statements without recommending a subgoal as the next course of action.

Referring to the sample dialogs at the beginning of the chapter, Dialog 1 was carried out in directive mode while Dialog 2 was carried out in declarative mode.[3] The computer verbally guided the user through every step in Dialog 1. On the other hand, in Dialog 2 the computer did not try to verbally verify many subgoals in order to provide cooperative utterances based on its perceptions of the user's focus during the problem solving process. Thus, variable initiative dialog complicates dialog processing for the subproblems of: (1) choosing the subgoals for completing the task, (2) moving between subdialogs, and (3) producing the expectations for user responses.

1.2.6 Integrated Behavior Via the Missing Axiom Theory

As will be seen in the next chapter, there has been a significant amount of important research on each target behavior. However, most of the work is based on an isolated study of an individual behavior, and there is a limited amount of work on integrating in one overall controlling process the mechanisms for obtaining each behavior. The Missing Axiom Theory mentioned in section 1.2.2 provides the linchpin for an integrated model. As illustrated in figure 1.1 and seen from the discussion in chapters 3 and 4, it does so in the following ways.

- Task and dialog are related via this theory through the theorems that define completion of task actions. Each theorem constitutes a subdialog, and the detection of a missing axiom in a proof attempt initiates the dialog interaction.

[3] While variable initiative behavior implies the ability to vary the initiative both between and within dialogs, our work emphasizes varying initiative between dialogs. A discussion of the difficulty in coherently varying initiative within a dialog is given in section 4.7.3.

- Within the above framework, maintenance of the user model as axioms provides a seamless interface for user model usage. This is done by determining the status of user knowledge as part of the action completion proofs. Missing user knowledge is detected as a missing axiom and may trigger a computer utterance to provide this knowledge.

- Expectations are produced according to the current dialog focus. This focus is provided by the action associated with the missing axiom that is triggering the language interaction. A record of these expectations for all active and previously active subdialogs can be used to determine when subdialog movement is occurring as well as assist in speech recognition.

- Finally, variable initiative behavior is enabled by augmenting the processing model to take into account the conflicting priorities of diverse user and computer goals and subdialog focus. Processing must vary as a function of the current dialog initiative mode.

1.3 Preliminary Study

Early in the development of the dialog processing model, many dialogs were collected and analyzed. The vast majority of these came from a study by Moody [Moo88] on the effects of restricted vocabulary size on discourse structure in spoken natural language dialog. Her results showed that human subjects could successfully adapt to a restricted vocabulary size. Furthermore, with increased expertise due to practice, subjects could become almost as efficient in completing the task as when subjects had an unrestricted vocabulary.

Moody's dialogs were collected using the "Wizard of Oz" paradigm, wherein a person simulates the computer in providing the needed expertise to human users for completing the circuit repair task. Because of the use of subjects in repeated trials, the simulations were conducted in such a way as to allow the human user to have control of the dialog at various times as their added experience permitted. Consequently, the dialogs were invaluable in validating many aspects of the dialog processing model, especially the work on variable initiative dialog, before it received the ultimate validation via testing the implemented system in experimental trials.

1.4 An Outline of the Book

Chapter 2 examines previous work on various problems in dialog processing. Chapter 3 presents the general dialog processing theory while chapter 4 gives the details of the computational model. Chapter 5 presents a theory of parsing spoken natural language that in conjunction with available dialog knowledge about possible user responses, enables a system to behave robustly in the presence of speech recognition errors. Chapter 6 describes the implemented system while chapters 7 and 8 present performance results based on experiments with the implemented system. Chapter 9 discusses an enhancement of the dialog processing model for verifying uncertain inputs. This enhancement was developed after the initial implementation was completed and the experiments were conducted. Finally, chapter 10 offers a concluding summary and critique of this research, highlighting ongoing and future areas of exploration.

Chapter 2

Foundational Work in Integrated Dialog Processing

Building a working spoken natural language dialog system is a complex challenge. It requires the integration of solutions to many of the important subproblems of natural language processing. This chapter discusses the foundations for a theory of integrated dialog processing, highlighting previous research efforts.

2.1 Problem Solving in an Interactive Environment

The traditional approach in AI for problem solving has been the planning of a complete solution. We claim that the interactive environment, especially one with variable initiative, renders such a strategy inadequate. A user with the initiative may not perform the task steps in the same order as those planned by the computer. They may even perform a different set of steps. Furthermore, there is always the possibility of miscommunication. Regardless of the source of complexity, the previously developed solution plan may be rendered unusable and must be redeveloped. This is noted by Korf [Kor87]:

> Ideally, the term planning applies to problem solving in a real-world environment where the agent may not have complete information about the world or cannot completely predict the effects of its actions. In that case, the agent goes through several iterations of planning a solution, executing the plan, and then replanning based on the perceived result of the solution.
>
> Most of the literature on planning, however, deals with problem solving with perfect information and prediction.

Wilkins [Wil84] also acknowledges this problem:

> In real-world domains, things do not always proceed as planned. Therefore, it is desirable to develop better execution-monitoring techniques and better capabilities to replan when things do not go as expected. This may involve planning for tests to verify that things are indeed going as expected.... The problem of replanning is also critical. In complex domains it becomes increasingly important to use as much as possible of the old plan, rather than to start all over when things go wrong.

Consequently, Wilkins adopts the strategy of producing a complete plan and revising it rather than reasoning in an incremental fashion. This may be satisfactory in some cases, but in a sufficently dynamic environment, developing a complete plan is often a waste of resources because the conditions under which the plan was developed may later be discovered to be inaccurate. This is particularly true in an interactive environment such as voice dialog where miscommunication can occur.

Recently there has been an interest in studying reasoning in dynamic environments where the conditions may change as the plan is being developed. Pollack and Ringuette [PR90] have constructed a system called Tileworld that consists of a simulated robot agent and a simulated environment that is dynamic and unpredictable. Their purpose is to experimentally evaluate the adequacy of various meta-level reasoning strategies in managing the explicit action planning that occurs during task performance.

Boddy and Dean [BD89] use a simulated world called *gridworld* that consists of a rectangular subset of the integer plane on which a robot courier must be scheduled to make as many pickups and deliveries as possible in a given amount of time. Since planning an optimal tour is in general a computationally expensive problem, they are interested in testing strategies that consider both the time for the robot to move from place to place as well as the planning time required for determining an optimal tour. Consequently, the optimal tour must minimize the sum of the traversal time together with the planning time required to construct the tour.

Although these two approaches do not consider an integration of such reasoning with NL dialog, they are consistent with our proposal for incrementally selecting the next task action to be performed without planning all the necessary remaining steps.

2.2 Language Use in a Problem-Solving Environment

2.2.1 The Missing Axiom Theory

Integrating dialog with the problem solving required for task completion necessitates a specification for the computational role of language. We propose the Missing Axiom Theory (section 1.2.2) that says language is used to acquire missing axioms needed for proving completion of actions. This view provides a practical computational paradigm for simulating human performance in a dialog. It seems clear that a human expert would focus responses on those parts of the task that the client is having difficulty with. Thus, although the Missing Axiom Theory may not be an accurate cognitive model of a person's thinking, it does lead to similar and effective surface behavior.

Quilici et al. [QDF88] use a similar approach in providing explanations for user misconceptions. User misconceptions are detected when the advisory system proves that it does not share a user's belief. Based on this proof, the system's explanation about the misconception must include a description of why it does not share the user's belief. Theorem-proving is also used in the process for computing an explanation for the user's erroneous belief. Thus, the missing user beliefs correspond to missing axioms that motivate the system's explanation.

Cohen and Jones [CJ89] also use a similar approach in selecting concepts to be discussed in responding to a user's query. Their domain is educational diagnosis. In this domain, the system is trying to assist a user in diagnosing the learning difficulties of a student. Such assistance involves developing a hypothesis about the cause of a student's learning difficulties along with suggesting a diagnostic procedure for verifying the hypothesis. The system's explanation takes into account missing user knowledge about the domain and/or the student.

Gerlach and Horacek [GH89] define rules for the use of language for a consultation system. The rules embody meta-knowledge about knowing facts and wanting goals as well as knowledge about the domain of discourse. Language is used when the system needs to inform the user that either: (1) a goal has been accomplished; or (2) a significant difference in the beliefs of the user and the system has been detected.

2.2.2 Speech Act Theory

In general, many researchers have proposed theories about the role or purpose of language. The origin of much of this research, including the Missing Axiom Theory, is *speech act* theory. As summarized in Allen [All87], speech act theory was developed based on the realization that statements can do more than just make assertions about the world. Statements can serve other functions. For example, when a minister says, "I now pronounce you husband and wife," at a wedding ceremony, the words act to change the state of the world. Another perspective is provided by Sadock [Sad90]:

> Speech act theory is concerned with providing an account of the fact that the use of expressions of natural language in context invariably involves the accomplishment of certain actions beyond the mere uttering (or writing, or telegraphing) itself. The context and the form of the utterance both enter into the equation. Holding the context constant and varying the utterance changes the accomplishments, and likewise holding the utterance constant and varying the context generally has profound effects on the actions that are performed.

Speech act theory, originally proposed by Austin [Aus62], is a development in the philosophy of language designed to explain the purpose of language in terms of the actions that speakers intend to perform by virtue of making an utterance. These types of actions are known as *illocutionary* acts. *Locutionary* acts are the actions associated with the physical production of the utterance. *Perlocutionary* acts are the effects that the utterance has on the hearer, independent of the effects the speaker intends. For example, if a police officer knocks on a door and asks the people inside if they have seen a certain criminal, the illocutionary act is to request information. However, if the criminal is hiding nearby, the utterance will also have the effect of warning and scaring the criminal although these were not the effects intended by the speaker. These are perlocutionary effects. The theories on the role of language that are of present concern involve the illocutionary acts of language.