Effects of Variable Initiative on Linguistic Behavior in Human-Computer Spoken Natural Language Dialogue

Ronnie W. Smith*
East Carolina University

Steven A. Gordon*
East Carolina University

This paper presents an analysis of the dialogue structure of actual human-computer interactions. The 141 dialogues analyzed were produced from experiments with a variable initiative spoken natural language dialogue system organized around the paradigm of the Missing Axiom Theory for language use. Results about utterance classification into subdialogues, frequency of user-initiated subdialogue transitions, regularity of subdialogue transitions, frequency of linguistic control shifts, and frequency of user-initiated error corrections are presented. These results indicate there are differences in user behavior and dialogue structure as a function of the computer's level of initiative. Furthermore, they provide evidence that a spoken natural language dialogue system must be capable of varying its level of initiative in order to facilitate effective interaction with users of varying levels of expertise and experience.

1. Modeling Human-Computer Dialogue

It is generally acknowledged that developing a successful computational model of interactive natural language (NL) dialogue requires extensive analysis of sample dialogues. Previous work has included analyses of (1) human-human dialogues in relevant task domains; (2) Wizard-of-Oz dialogues in which a human (the Wizard) simulates the role of the computer as a way of testing out an initial model; and (3) human-computer dialogues based on initial implementations of computational models. Each of these dialogue types has advantages as a model for system building, in terms of the relevance of the data to the final model. However, each also has particular disadvantages when researchers attempt to generalize from the findings of previous work.

For example, much analysis of human-human interactions has been done, such as Walker and Whittaker's (1990) analysis of mixed initiative in dialogue, or Oviatt and Cohen's (1991) comparison of interactive and non-interactive spoken modalities. Analyses of human-human dialogues are a good basis for an initial task model and a lexicon, but it is difficult to determine which aspects of these analyses will generalize to human-computer dialogues and which ones will not. Fraser and Gilbert (1991) note that "although it is certainly better to rely on analyses of human-human interactions than to rely on intuitions alone, the fact remains that human-human interactions are not the same as human-computer interactions and it would be surprising if they followed precisely the same rules" (p. 81). In addition, Oviatt and Cohen (1991) say that "... to model discourse accurately for interactive systems further research clearly will be needed on the extent to which human-computer speech differs from that between humans. At present, there is no well developed model of human-machine communi-

* Department of Mathematics, Greenville, NC 27858, USA. First author's e-mail: rws@cs.ecu.edu

cation ..." (p. 323). The dilemma of researchers is nicely summarized by Fraser and Gilbert: "The designer is caught in a vicious circle—it is necessary to know the characteristics of dialogues between people and automata in order to be able to build the system, but it is impossible to know what such dialogues would be like until such a system has been built" (p. 81).

Wizard-of-Oz (WOZ) dialogues result from an experimental technique that is one way of addressing this dilemma. In this methodology, human subjects are told they are interacting with a computer when they are really interacting with another human (the Wizard) who simulates the performance of the computer system. In some simulations (e.g., Whittaker and Stenton [1989]), the Wizard simulates the entire system while in other cases (e.g., Dahlbäck, Jönsson, and Ahrenberg [1993]), the Wizard makes use of partially implemented systems to assist in responding. Consequently, many initial models can be prototyped and tested before implementation, and researchers need not have a fully developed natural language interface. As other researchers have noted (Whittaker and Stenton 1989; Dahlbäck, Jönsson, and Ahrenberg 1993; Fraser and Gilbert 1991), when the WOZ simulations are convincing, they obtain data that are a more accurate predictor of actual human-computer interaction than human-human dialogues because speakers adapt to the perceived characteristics of their conversational partners. Consequently, WOZ studies can provide an indication of the types of adaptations that humans will make in human-computer interaction. WOZ studies such as the ones cited above have been particularly useful in obtaining data on discourse structure and contextual references. The WOZ study of Moody (1988) on the effects of restricted vocabulary on interactive spoken dialogue provided the data that influenced the development of our own system.

While much knowledge can be gained from WOZ studies, they are not an adequate means of studying all elements of human-computer natural language dialogue. A simulation is feasible as long as humans can use their own problem-solving skills in carrying out the simulation, but when it requires mimicking a proposed algorithm, the WOZ technique becomes impractical. For example, it is difficult to simulate and test the computer's error recovery strategies for speech recognition or natural language understanding errors, because the natural language understanding of the computer is only a simulation. If we wish to test an actual computational model for natural language processing, its complexity demands the construction of a computer program to execute it. Furthermore, an important feature of dialogue that is difficult to simulate via the WOZ paradigm is that of initiative. Depending on the interaction environment, dialogue initiative may reside with the computer, with the user, or may change during the interaction. Lacking any formal models of initiative, it would be very difficult for a Wizard to accurately simulate the response patterns a computerized conversational participant would produce in a mixed-initiative dialogue for a nontrivial domain that would be consistent from subject to subject.

Unfortunately, we can also have difficulties generalizing from analyses of human-computer dialogues, because parameters of the particular system with which the dialogues were collected may have significantly affected the resulting dialogues. For example, if a particular system is always run with a particular speech recognizer, it may be difficult to determine what the outcome would have been with a better speech recognizer. Similarly, most human-computer dialogues are collected from systems with a particular dialogue model. Since it is well known that users adapt to the system, it will be unclear how the results from a particular set of human-computer dialogues generalize to a model of interaction based on a different dialogue model.

This paper reports work that attempts to address both of these dilemmas through the analysis of human-computer dialogues collected in an environment in which

aspects of the system are parameterizable. We have built an integrated dialogue-processing system, the Circuit Fix-It Shop, which is parameterized for a key system behavior: initiative.[1] We have tested the system in 141 dialogues totaling 2,840 user utterances while varying levels of system initiative. The paper discusses our model of initiative and presents quantitative results from an analysis of our corpus on the effect of the computer's level of initiative on aspects of human-computer dialogue structure such as (1) utterance classification into subdialogues, (2) frequency of user-initiated subdialogue transitions, (3) regularity of subdialogue transitions, (4) frequency of linguistic control shifts, and (5) frequency of user-initiated error corrections. The results indicate there are differences in user behavior and dialogue structure as a function of the computer's level of initiative. Furthermore, they provide evidence that a spoken natural language dialogue system must be capable of varying its level of initiative in order to facilitate effective interaction with users of varying levels of expertise and experience.

2. A Theory of Variable Initiative Dialogue

In this section we review a theory presented in Smith and Hipp (1994). It is important to note that our focus is on task-oriented dialogues, that is, dialogues whose purpose is to discuss a task whose completion is being carried out at the same time as the dialogue. Consequently during the discussion, we make the distinction between **linguistic goals** and **task goals**. Linguistic goals relate to speaker intentions in making statements (e.g., to inform, command, or request), while task goals relate to specific actions that need to be carried out in the domain of interest in order to complete the task (e.g., performing a voltage measurement). As will be seen from the discussion, we take the approach that task initiative is assigned to the participant whose current task goals have priority, and the purpose of dialogue initiative is to indicate who has the task initiative.

2.1 Defining Variable Initiative and Dialogue Mode

Variable initiative dialogue is dialogue in which: (1) either dialogue participant can have control of the dialogue, (2) control can vary between participants during the dialogue, and (3) intermediate levels of control are allowed.

A variable initiative dialogue system contrasts with other NL dialogue systems such as those described in Section 3.1 in which the dialogue is either purely user-controlled or purely computer-controlled. In user-controlled dialogue systems the computer acts as a passive agent responding to user queries. Question-answering systems are examples of user-controlled dialogue systems.[2] In computer-controlled dialogue systems, the user is totally dependent on the computer for accomplishment of the task.

The need for variable initiative dialogue arises because at some points during task completion a user may have sufficient knowledge to take control of the dialogue and accomplish several goals without much computer assistance while at other times, a user may need detailed assistance. Thus, user initiative is characterized by giving priority to the user's goals of carrying out steps uninterrupted while computer initiative is characterized by giving priority to the specific goals of the computer. In general, we have observed that the level of initiative that the computer has in the dialogue is

1 Smith and Hipp (1994) presents details about the overall computational model that forms the basis of the system.

2 While some question-answering systems can initiate clarifications to disambiguate user queries, the user remains in control of the overall interaction.

primarily reflected in the degree to which it allows the user to interrupt the current subdialogue in order to discuss another task goal. When the user has control, the interrupt is allowed, but when the computer has control it is not.[3] However, initiative is not an all-or-nothing control mechanism. Either the user or the computer may have the initiative without having complete control of the dialogue. Based on these observations, four dialogue **modes** were identified that characterize the level of initiative that the computer can have in a dialogue. These are described below.

1. Directive: The computer has complete dialogue control. It recommends a task goal for completion and will use whatever dialogue is necessary to complete this goal. No interruptions to other subdialogues are allowed.
2. Suggestive: The computer still has dialogue control, but not as strongly. The computer will recommend a task goal for completion, but will allow minor interruptions to closely related subdialogues.[4]
3. Declarative: The user has dialogue control and can interrupt to any desired subdialogue at any time. However, the computer is free to mention relevant facts as a response to the user's statements.
4. Passive: The user has complete dialogue control. Consequently, the computer will provide domain information only as a direct response to a user question.

2.2 Response Formulation in Variable Initiative Dialogue

Since the degree of interruptibility allowed by the computer increases from directive to passive mode, dialogue mode has a critical effect on the computer's choice of response. As illustrated in Figure 1, response topic selection is a function of the computer's goal, the user focus (i.e., the task goal that the computer believes the user currently wants completed), and the dialogue mode. When the user focus differs from the computer's goal (i.e., an interrupt), the dialogue mode becomes the decisive factor in the selection process, as described in Figure 2. The mechanics of the response selection algorithm can be illustrated via the following situation: suppose that the user initially states, "the light is off," and suppose the computer knows that in order for the light to be lit, the switch must be turned up (i.e., **state(switch,up)** $\Rightarrow$ **state(light,on)**). Consequently, the computer goal of highest priority is to put the switch up, while the user focus is assumed to be on the light.[5] The selection process as a function of mode would be as follows:

1. Directive: User focus is ignored with no interrupts permitted. The selected goal is to put the switch up.
2. Suggestive: User focus is seen to be related to the computer goal via the domain fact relating the switch and the light. The selected goal is to "observe the switch position when the light is off."

3 Note that we do not consider clarification subdialogues to be interrupts, as the overall task goal remains unchanged.

4 Subdialogues about different task goals are considered closely related if the different objects of interest share a sufficiently close common ancestor in the domain-knowledge hierarchy.

5 A complete plan recognition process for inferring the user's exact goal has tended to be a very costly computational process and not feasible in a system designed for real-time interaction. Consequently, our system uses the user focus as determined from the previous utterance as a basis for its beliefs about the user's current goals.

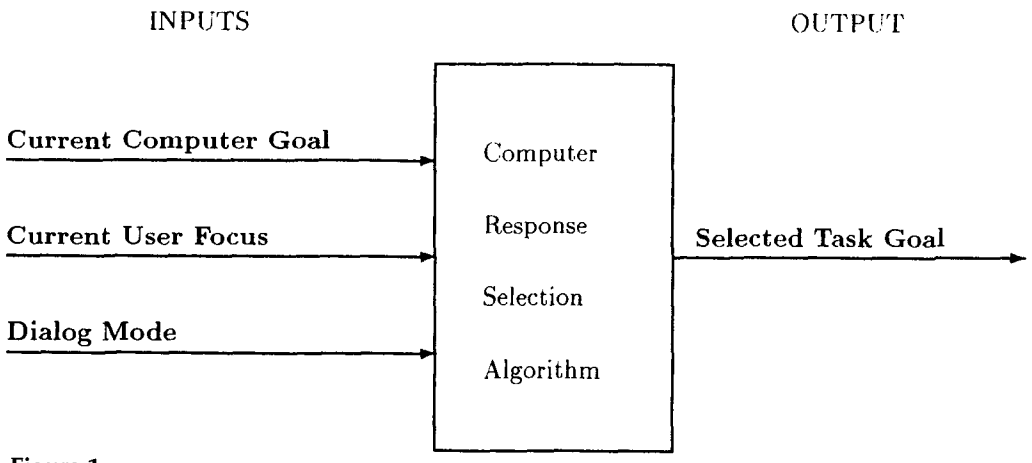

Figure 1
Flow diagram of the computer response selection process.

```
IF Mode = directive THEN
        select the computer goal without any regard for the user's focus
ELSE IF Mode = suggestive THEN
        search the domain knowledge hierarchy for a common relationship
            between the computer goal and the user focus
        IF such a relationship exists THEN
           select this as the next goal
        ELSE
           select the computer's original goal
ELSE IF Mode = declarative THEN
        search the domain knowledge hierarchy for a common relationship
            between the computer goal and the user focus
        IF such a relationship exists THEN
           select as the next goal that the user learn about this
               relationship
        ELSE
           select as the next goal an uncommunicated fact relevant to
               the user focus
ELSE IF Mode = passive THEN
        select as a goal that the user learn the computer has processed
            the user's last utterance
```

Figure 2
Computer response selection algorithm.

3. Declarative: User now has control. Consequently, the selected goal must be a relevant fact. The previous goal is converted to "user learn that the light is on when the switch is up."
4. Passive: User has complete control. Computer simply acknowledges processing the last user utterance.

This response selection process has been implemented as part of the previously mentioned Circuit Fix-It Shop. The two dialogues of Figure 3, obtained from actual usage of the implemented system, illustrate differences between the two modes in

which the system was experimentally evaluated: directive and declarative. Note the following phenomena in these dialogues.

1. In the directive mode dialogue, the subject is performing task goals under the close guidance of the computer. There is language interaction about each task goal.[6]
2. In the declarative mode dialogue, the subject independently carries out several task goals, known to be necessary, without any interaction. By allowing the user to arbitrarily change subdialogues, the computer is able to provide relevant assistance when a potential problem is reported without requiring language interaction for the task goals already completed.

A variable initiative dialogue system is just the first step toward the more important objective of a **mixed-initiative** dialogue system. In a mixed-initiative interaction, initiative can vary between the participants throughout the dialogue. Given that our first priority in experimentally evaluating the system was to demonstrate that behavior varied as a function of initiative, it was necessary to fix the level of initiative for the duration of a session. We next review the work of others who have examined issues in mixed-initiative interaction.

2.3 Prior Theoretical Work on Mixed-Initiative

Whittaker and Stenton (1988) propose a definition for dialogue control based on the utterance type of the speaker (question, assertion, command, or prompt) as follows:

- Question: The speaker has control unless the question directly followed a question or command by the other conversant.
- Assertion: The speaker has control unless the assertion was a response to a question.
- Command: The speaker has control.
- Prompt: The listener has control because the speaker is abdicating control.

Walker and Whittaker (1990) apply these dialogue control rules to analyze how initiative affects discourse structure. Their analysis distinguishes between advisory dialogues and task-oriented dialogues, but they do not allow for the possibility that the novice in a task-oriented dialogue can gain knowledge over time and want more control of the dialogue. Their key observations include the following:

- Transfer of control is often a collaborative phenomenon. Since a noncontrolling participant has the option of seizing control at any moment, then the controlling participant must have control because the noncontroller allows it.

6 For brevity, dialogue 1 represents one of the simplest directive mode interactions that could occur. For more complex examples, the transcripts of the dialogues collected during the experiment are available by anonymous FTP. Contact the first author for details.

```
   Dialogue 1 (Directive Mode)

 1. C: This is the Circuit Fix-It Shop.  How may I help you?
 2. U: I want to fix a circuit.
 3. C: What is the ID of the circuit?
 4. U: RS111.
 5. C: I am familiar with that circuit.  The LED is supposed to be displaying
       alternately flashing one and seven.
 6. U: The LED is off.
 7. C: What is the switch at when the LED is off?
 8. U: The switch is down.
 9. C: Put the switch up.
10. U: Okay.
11. C: Put the knob to one zero.
12. U: Okay.
13. C: What is the LED displaying?
14. U: LED is displaying only a flashing seven.
15. C: Is there a wire between connector eight four and connector nine nine?
16. U: No.
17. C: Add a wire between connector eight four and connector nine nine.
18. U: Done.
19. C: What is the LED displaying?
20. U: Alternately flashing one and seven.
21. C: Is the one on the LED displaying for a longer period of time?
22. U: No.
23. C: Is the seven on the LED displaying for a longer period of time?
24. U: No.
25. C: Put the knob to zero.
26. U: Okay.
27. C: What is the LED displaying?
28. U: Alternately displaying one and seven.
29. C: Is the one on the LED displaying for a longer period of time?
30. U: Yes.
31. C: Put the switch down.
32. U: The switch is down.
33. C: What is the LED displaying?
34. U: Nothing.
35. C: The circuit is working correctly.  Good-bye.

*******************************************************************************

    Dialogue 2 (Declarative Mode)

 1. C: This is the Circuit Fix-It shop.  How may I help you?
 2. U: RS111 is not working.
 3. C: I am familiar with that circuit.  The LED is supposed to be displaying
       alternately flashing one and seven.

       NOTE:  Subject put the switch up and set the knob to one zero, observed
              the LED display, and noted the potential problem without
              requiring any assistance from the computer.

 4. U: There is no wire from connector nine nine to connector eight four.
 5. C: There is supposed to be a wire between connector nine nine and
       connector eight four.

       NOTE:  Subject then added the missing wire, and manually performed all
              necessary checks to verify the circuit was functioning properly.

 6. U: The circuit is working.
 7. C: Glad to have been of assistance.  Good-bye.

*******************************************************************************
```

Figure 3
Sample dialogues: directive and declarative.

- The expert retains more control in the task-oriented dialogues, but there are still occasional control changes when the novice has to describe problems that are occurring while completing the task.
- Summaries are more frequent in advisory dialogues due to the need for both participants to verify that they do share the mutual beliefs needed to develop the necessary plan.

In Section 6.5 we investigate the relationship of this notion of dialogue control based on linguistic goals to our task goal notion of control.

Kitano and Van Ess-Dykema (1991) extend the plan recognition model of Litman and Allen (1987) to consider mixed-initiative dialogue. Their key insight is the observation that the two participants may have different domain plans that can be activated at any point in the dialogue. Thus, there are speaker-specific plans instead of simply joint plans as in the Litman and Allen model. This separation of plans permits greater flexibility in the plan recognition process. Furthermore, they extend the initiative control rules proposed by Whittaker and Stenton to consider the utterance content by observing that a speaker has control when the speaker makes an utterance relevant to his or her speaker-specific domain plan. Although they do not consider a computational model for participating in mixed-initiative dialogues, their observation that there are speaker-specific plans or goals underlies the model that we propose.

2.4 Theory Evaluation

While WOZ simulation of directive and passive modes is feasible, the requirements for algorithmically determining the relationship between user focus and the computer goal make WOZ simulations of suggestive and declarative modes very difficult, especially given the fast response time necessary for spoken interaction. Before the construction of the Circuit Fix-It Shop, Moody (1988) conducted a Wizard-of-Oz study on the effects of restricted vocabulary on interactive spoken dialogue. Her data were the basis for the formulation of the experimental Circuit Fix-It Shop system. Although she attempted to acquire information concerning user behavior when users were given the initiative, she was unable to provide much information because her subjects did not interact with the system enough to evolve from novices to experts. Her attempts to yield the initiative to users still led to statements that guided users step-by-step through the task. By direct testing of a computer system that implements our proposed model of variable initiative dialogue, we could more rigorously control the system performance and more easily run repeated tests with subjects and allow them to gain task expertise. Simultaneously, we could more readily monitor the effects of the change in initiative setting while holding other system features constant.

In testing our theory of variable initiative dialogue, there were two main types of phenomena we wished to examine: (1) general aspects of task efficiency, such as time to completion and number of utterances spoken; and (2) the nature of the dialogue structure. Results on task efficiency are reported in detail in Smith and Hipp (1994) and are briefly reviewed in Section 6.1. The primary contribution of this paper is to present an analysis of how the dialogue structure varies according to the computer's level of initiative. After reviewing some details about the overall dialogue-processing model and its implementation, in Section 3, and a review of the experimental environment, in Section 4, the remainder of the paper focuses on the results of this analysis, a review of some related analyses, and some concluding remarks about the usefulness of the analysis and the role of experimental natural language dialogue systems in modeling human-computer dialogue.

3. Dialogue-Processing Model: An Integrated Approach

3.1 Motivation and Overview

Most prior work on natural language dialogue has either focused on individual subproblems such as quantification, presuppositions, ellipsis, anaphoric reference, and user modeling, or else focused on dialogue-processing issues in database query applications. Examples of such dialogue systems are described in Allen, Frisch, and Litman (1982), Bobrow et al. (1977), Carberry (1988), Frederking (1988), Hafner and Godden (1985), Hendrix et al. (1978), Hoeppner et al. (1983), Jullien and Marty (1989), Kaplan (1982), Levine (1990), Peckham (1991), Seneff (1992), Waltz (1978), Wilensky et al. (1988), Young et al. (1989), and Young and Proctor (1989). However, there has been little work on integrating the various aspects of dialogue processing into a unified whole (exceptions are Allen et al. [1995] and Young et al. [1989]). Consequently, we developed a dialogue-processing model for task-oriented dialogues that when implemented in an electronic repair domain exhibits a number of important behaviors including: (1) problem-solving; (2) coherent subdialogue movement; (3) user model usage; (4) expectation usage; and (5) variable initiative behavior. We summarize the key features of the model below.

- Theorem proving is used as the reasoning mechanism for determining when task goals are completed.
- Consequently, the purpose for language during the dialogue is to acquire the missing axioms needed for proving task goal completion (i.e., The Missing Axiom Theory [Smith 1992]).
- User model information is maintained as a set of axioms acquired from inferences based on user input. The axioms may then be used by the theorem prover.
- Finally, integration of theorems, the utterances relevant to these theorems, and the expectations for responses that supply missing axioms yields a constructive method for creating and using a discourse model first proposed by Grosz and Sidner (1986), but for which they did not offer a method of dynamic construction during the course of a dialogue.

Furthermore, the model enables the system to engage in variable initiative dialogue as outlined in Section 2. The interested reader is referred to Smith, Hipp, and Biermann (1995) for further details about the overall model.

3.2 System Implementation

We constructed the Circuit Fix-It Shop based on the details of our dialogue-processing model. The system was originally implemented on a Sun 4 workstation with the majority of the code written in Quintus Prolog and the parser in C. The system assists users in the repair of a Radio Shack 160 in One Electronic Project Kit. The system can detect errors caused by missing wires as well as a dead battery.

Speech recognition is performed by a Verbex 6000 running on an IBM PC. To improve speech recognition performance, we restrict the vocabulary to 125 words. A DECtalk DTCO1 text-to-speech converter is used to provide spoken output by the computer.

An important feature of any spoken natural language dialogue system is the ability to perform robust parsing. Spoken inputs are frequently ungrammatical but must still

be interpreted correctly. The main source of ungrammatical inputs in our experiments was the misrecognition of the user's input. An error-correcting parser was developed that finds the minimal cost set of insertions, deletions, and substitutions to transform the input into grammatical input (Smith and Hipp 1994). During our formal experiment, the system was able to find the correct meaning for 81.5% of the more than 2,800 input utterances even though only 50% of these inputs were correctly recognized word for word. An overview of the experimental design is presented next.

4. Experimental Design

The experimental design is discussed in great detail in Smith and Hipp (1994) and Smith (1991). Here we present an overview of the experiment sufficient for understanding the environment in which the data were collected.

4.1 Subject Pool

The eight subjects were Duke University undergraduates who met the following criteria.

- They had demonstrated problem-solving skills by having successfully completed one computer science course and had taken or were taking another.
- They did not have excessive familiarity with AI and natural language processing. In particular, they had not taken a class in AI and they had not interacted with a natural language system.
- None were majoring in electrical engineering. Such individuals could probably fix the circuit without any assistance.

The subject pool consisted of six male and two female subjects. In addition, two pilot subjects, one female and one male, were run using the proposed experimental design before the formal experiment began.

4.2 Session Overview and Problem Selection

Subjects participated in the experiment in three sessions. The first and third sessions occurred a week apart, and the second session normally occurred three or four days after the first session.[7] The first session consisted of: (1) the primary speech training, lasting approximately 60 to 75 minutes; (2) approximately 20 minutes of instruction on using the system; and (3) practice using the system by attempting to solve four "warmup" problems with the system operating in directive mode, the mode where the computer has maximal control. A maximum of two and one-half hours was spent on the first session. The second and third sessions each consisted of: (1) review work with the speech recognizer; (2) a review of the instructions; and (3) usage of the system on up to 10 problems depending on how rapidly the problems were solved. One group of subjects worked with the system in directive mode during the second session and in declarative mode during the third session while the other group worked with the same modes, but in opposite sessions. The time allowed for the second and third sessions was two hours each.

7 The only exception was the last subject, where the second session occurred two days after the first session, and the third session occurred one week after the second session.

The particular circuit being repaired is supposed to cause the LED to alternately display a 1 and a 7, and the implemented domain problem-solving component could detect errors caused by missing wires as well as a dead battery. The basic debugging process consists of the following steps:

1. Determine if the LED display is correct.
2. If it is not correct, perform zero or more diagnostic steps to further isolate the problem. Possible diagnostic steps are voltage measurements or an LED observation under a different physical configuration of the circuit.
3. Check for the absence of one or more wires until a missing wire is identified.

The wires are attached to metal spring-like connectors, which are identified by numbers on the circuit board. Thus, a wire is identified by the numbers of the two connectors to which it is connected. In order to balance the difficulty of the problems between the second and third sessions, the wires were classified according to the number and type of diagnostic steps required to detect the error. Based on this classification, the assignment of missing wires to problems in each session was made as follows:

- Four wires were used in the four warmup problems of the first session.
- From a set of 10 other wires, 5 were used for the first five problems of session 2 and the other 5 were used for the first five problems of session 3. Each of these problems was balanced for difficulty. For example, problem 1 of both sessions was a power subcircuit problem, while problem 5 of both sessions was an LED subcircuit problem. Problems 2 through 4 were similarly balanced.
- Problems 6 through 8 of sessions 2 and 3 consisted of 2 missing wires for each problem. The 2 missing wires were selected from the 5 missing wires used during the first five problems of the session. Each of problems 6 through 8 differed by one missing wire. These problems were also balanced for difficulty.
- Problems 9 and 10 of each session consisted of a missing wire that was also used during the warmup problems of session 1. Each of these 4 wires was assigned to a different problem. Consequently, sessions 2 and 3 are balanced for difficulty only through the first eight problems.

4.3 Experimental Setup

Figure 4 provides a rough sketch of the room layout. The subject was seated facing the desk containing the circuit board. Communication with the speech recognizer was performed through a telephone handset. The experimenter was seated in front of the computer console. Thus, the subject's back was to the experimenter. The experimenter had a copy of the raw data form for the session, a copy of the word list, and a guide describing the allowed experimenter interaction with the subject. Data collection mechanisms consisted of the following:

1. Automatic logging of the words received from the speech recognizer (subject input) and the words sent to the DECtalk (computer output).

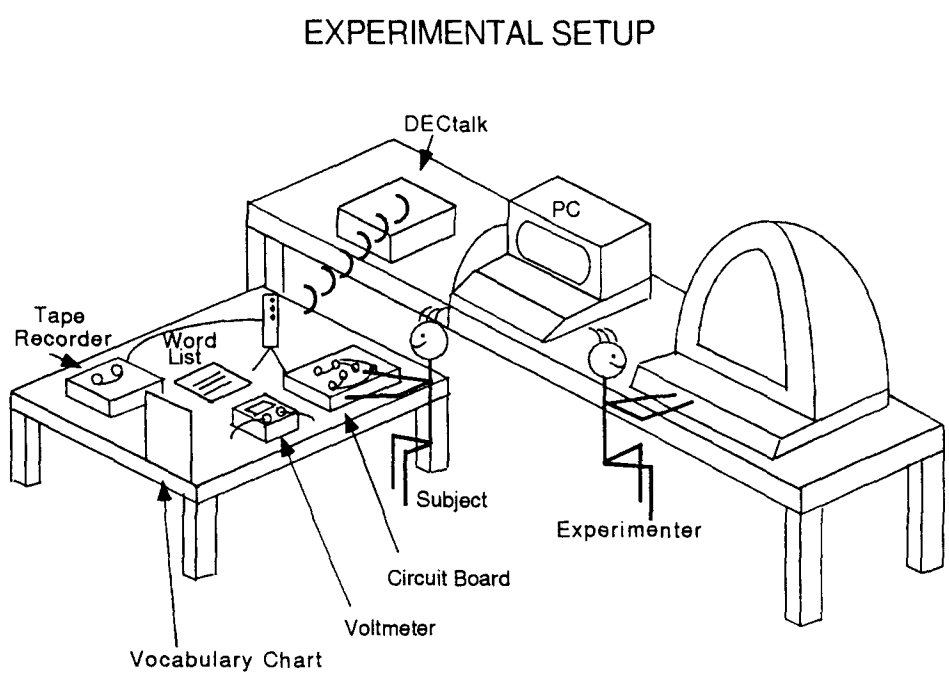

Figure 4
Room setup.

This logging information included the time the words were received or sent. In addition, time information was recorded for when the parser finished its processing of the input and when the computation of the input interpretation was complete.

2. The interaction was tape recorded in order to make a transcript that included the actual words used by the subject and the interactions that occurred between the subject and the experimenter.

3. The experimenter made notes about the interaction on the raw data form as well as marked occurrences of subject-experimenter interaction according to the category into which the interaction could be classified. In order to assist the experimenter in determining when a misrecognition occurred, the experimenter monitored the file where automatic logging occurred.

4.4 Experimenter Interaction

An important issue in experiments such as this, as has been observed elsewhere (Biermann, Fineman, and Heidlage 1992), is the problem of giving the subject sufficient error messages to enable satisfactory progress. One major source of difficulty in this experiment were misrecognitions by the Verbex speech recognizer. These miscommunications created various problems for the dialogue interaction, ranging from repetitive dialogue to experimenter intervention to occasional failure of the dialogue. Whenever a serious misrecognition caused the computer to interpret the utterance in a way that contradicted what was meant, the experimenter was allowed to (1) tell the subject that a misrecognition had occurred, and (2) tell the subject the interpretation made by the computer, but *could say nothing else*. For example, when one subject said, "the circuit is

working," the speech recognizer returned the words "faster it is working." This was interpreted as the phrase *faster*. Consequently, the experimenter told the subject, "Due to misrecognition, your words came out as *faster*." It is important to note that when an utterance was misunderstood, the experimenter did not tell the subject what to do, but merely described what happened. In this way, the interaction was restricted to being between the computer and the subject as much as possible, given the quality of commercial, real-time, continuous speech recognition devices at the time of the experiment. Such error messages from the experimenter occurred, on average, once every 15 user-utterances throughout the experiment.

The other main source of difficulty in using the system was the enforcement of the single utterance, turn-taking protocol of the interaction. This required the user to signal the beginning of an utterance by speaking the sentinel word *verbie* and end the utterance with the word *over*. Users would sometimes forget to use the sentinel words or else would not wait for the system's response that would occasionally be delayed up to 30 seconds (normal response time was 5 to 10 seconds). In cases where the interaction protocol was violated, the experimenter would issue a warning statement such as, "Please be patient. The system is taking a long time to respond," or "Please remember to start utterances with *verbie*." These types of experimenter interactions occurred, on average, once every 33 user-utterances.

4.5 The Nature of the Spoken Dialogue

The limitations of real-time continuous speech recognition at the time of the experiments had an impact on the nature of the spoken human-computer interaction that was observed in comparison to what might be expected in a spoken human-human interaction. In particular, the restrictive 125-word vocabulary meant that speech repairs and disfluencies that are prevalent in human-human spoken interaction and an important area of study (Oviatt 1995; Heeman and Allen 1994) could not be processed by the system. Whenever a person misspoke, they could start over by issuing the sentinel word *cancel*, rather than *over* at the end of their utterance. To prevent this from happening often, subjects were instructed at the start of their participation to plan their utterance completely before speaking. Consequently, there were only 11 cancels issued in the production of the 2,840 user-utterances. Furthermore, in exit interviews conducted after they had completed participation, none of the subjects indicated any difficulty with, or dislike of, planning utterances in advance.

To summarize, the results in Section 6 on the structure of spoken natural language dialogue are based for the most part on planned speech, a consequence of the technological limitations of speech recognizers at the time. Nevertheless, we believe it represents the first widely reported and analyzed spoken human-computer co-operative problem-solving dialogue, and that it is representative of such dialogue for the forseeable future.

5. Classifying Dialogue Utterances

5.1 Major Subdialogues in Repair Assistance

For task-oriented dialogues Grosz (1978) has noted that *the structure of a dialogue mirrors the structure of the underlying task*. Moody (1988) conducted a Wizard-of-Oz study on the effects of restricted vocabulary on interactive spoken dialogue. Her data were the basis for the formulation of the experimental Circuit Fix-It Shop system. For repair

Table 1
Utterance classification into major subdialogues.

Subdialogue Type	Directive Dialogue Utterances	Declarative Dialogue Utterances
Introduction	1–4	1–2
Assessment	5–14	3
Diagnosis	15–16	4–5
Repair	17–18	—
Test	19–35	6–7

tasks, she identified five primary task subdialogues:

- Introduction Subdialogue (I): Establish the purpose of the task (e.g., to fix the circuit with ID number RS111).
- Assessment Subdialogue (A): Establish the current behavior.
- Diagnosis Subdialogue (D): Establish the cause for the errant behavior.
- Repair Subdialogue (R): Establish that the correction for the errant behavior has been made.
- Test Subdialogue (T): Establish that the behavior is now correct.

Table 1 shows the classification into the various subdialogues of the utterances from the sample dialogues of Figure 3.

5.2 Subdialogue Transition

Another important aspect of the dialogue structure is the nature of the transitions between subdialogues. The model we present is derived from Moody's (1988) study, mentioned above. In the absence of errors in completing task actions, the natural transition from subdialogue to subdialogue is described by the following regular expression:

$$I^+A^+(D^+R^*T^+)^nF$$

where "+" denotes that one or more utterances will be spoken in the given subdialogue, "*" denotes that zero or more utterances will be spoken in the given subdialogue, and n represents the number of individual repairs in the problem.[8] The letters correspond to the abbreviations given in Section 5.1, and F represents the finished state (i.e., completion of the dialogue). This transition model is also depicted in the finite-state network of Figure 5. For clarity, loop arcs (i.e., transitions from a subdialogue back into itself) are omitted. We see from this model that dialogues normally begin with the Introduction and Assessment phases. Once the errant system behavior is described, the dialogue goes through one or more cycles of Diagnosis, Repair, and Test, until the system behavior is correct.

8 In our domain, n represents the number of missing wires in the problem. For example, when there are two missing wires, the first *DRT* iteration will cause one missing wire to be added, but the Test phase will show that the circuit is still not working. A second *DRT* iteration is required to detect and add the missing wire that completes the repair.

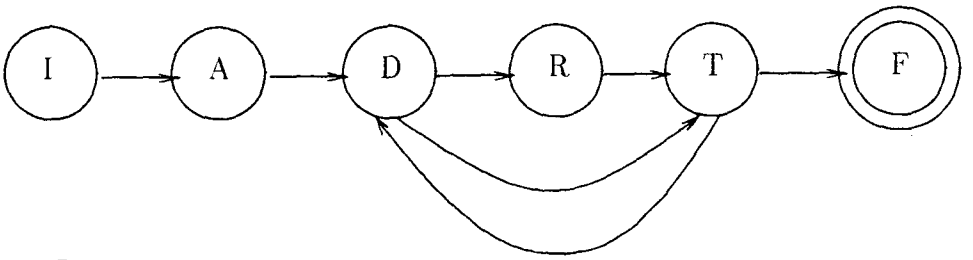

Figure 5
Subdialogue transition as a finite-state network.

This model was helpful in classifying each utterance into the appropriate subdialogue. As discussed in Section 6.4, however, not all dialogues followed this model, due to user initiative and dialogue miscommunication. Nevertheless, it provides a good first approximation of the nature of subdialogue movement.

5.3 Transcript Coding

The two authors each coded the transcripts independently. Every utterance (those spoken by the computer as well as those spoken by the human subject) was classified into one of these five subdialogue categories, according to two perspectives: the speaker's perspective (i.e., the task subdialogue that the speaker of the utterance believed was relevant to the statement) and the global perspective (i.e., the task subdialogue that is relevant to the utterance, based on omniscient knowledge of the task status). Normally these were the same, but not always. In situations where the user carried out a repair without explicitly notifying the computer, the computer might think the task was still in one phase, when the user had actually moved the task into another phase. In the results to be presented, the current subdialogue is based on global, rather than speaker, perspective. Overall, there was a difference between speaker and global perspective in 6.7% of the declarative mode utterances and in 1.7% of the directive mode utterances.

5.4 Coding Reliability

The two authors compared their coding results as the transcripts for each one of the eight subjects were completed, in order to resolve differences and, hopefully, improve agreement as more transcripts were coded. The first author was a principal designer of the system, while the second author had only watched a videotape of the system in operation and read some of the previous papers about the project. Consequently, many of the initial disagreements in coding were due to a lack of familiarity with what transpired during the experiment. For example, in situations where the Repair subdialogue was not explicitly verbalized, it was not clear whether subsequent descriptions of the circuit behavior indicated that the current subdialogue was Test or Assessment. Proper coding in these situations required familiarity with what had actually occurred during the experiment, familiarity that only the first author had. For all dialogues, initial interrater agreement on both speaker and global perspective of the current subdialogue was 87.2%. That is, for 12.8% of the utterances, there was a disagreement between the coders over either speaker perspective of the current subdialogue, global perspective, or both. The kappa coefficient (Isard and Carletta 1995) for the level of agreement is 0.82. When the coding process was completed, all discrepancies were resolved to the satisfaction of both authors.

6. Results

6.1 Data Inclusion and Statistical Analysis

Subjects attempted a total of 141 dialogues, of which 118 or 84% were completed successfully.[9] The average speech rate by subjects was 2.9 sentences per minute, and the average task completion time for successful dialogues was 6.5 minutes. The system had an average response time of 8.1 seconds during the formal experiment. Later, a faster parsing algorithm was implemented and the system was ported to a SPARC II workstation from the Sun 4 used during the experiment. During test dialogues using the enhanced system, average response time was 2.2 seconds.

In general, differences in user behavior depending on the level of computer initiative were observed. When the computer operated in declarative mode—yielding the initiative to human users, who could then take advantage of their acquired expertise—the dialogues:

- were completed faster (4.5 minutes versus 8.5 minutes).
- had fewer user-utterances per dialogue (10.7 versus 27.6).
- had users speaking longer utterances (63% of the user-utterances were multiword versus 40% in directive mode).

While users given the initiative in the final session were somewhat more efficient at completing the dialogues than users given the initiative in the second session (completing dialogues approximately 1.5 minutes faster and speaking on average 2.7 fewer utterances), the large standard deviations, which ranged from 50% to 90% of the associated sample means, and the small number of subjects tested indicate that we should use caution in generalizing from our results.

Unless explicitly noted, the results on human subjects' linguistic behavior that will be reported throughout this section are based only on the 118 dialogues that were successfully completed. While the 23 incomplete dialogues also contain interesting phenomena, we chose to focus the analysis on the completed dialogues, as they represent the linguistic record of successful interactions with the system. In reality, there are only slight differences in the results when the unsuccessful dialogues are included. Furthermore, a valid statistical analysis could only be performed on the completed dialogues. Reporting data values from only the successful dialogues maintains consistency with the reported statistical values.

6.2 Utterance Classification into Subdialogues

6.2.1 Hypotheses. For users to take the initiative in the task domain, they must have some expertise in the domain. Once this expertise is gained, and the computer yields task control to the human user, it is expected that users will exploit the situation to restrict the dialogue to specific issues of interest. Presumably, such users have substantial knowledge about the general behavior of the circuit, how to determine when

9 Due to time constraints, not all subjects were able to attempt all possible dialogues. Only three of the eight subjects successfully completed all possible dialogues. Of the 23 dialogues not completed, 22 were terminated prematurely due to excessive time being spent on the dialogue. Misunderstandings due to misrecognition were the cause in 13 of these failures. Misunderstandings due to inadequate grammar coverage occurred in 3 of the failures. In 4 of the failures, the subject misconnected a wire. In one failure there was confusion by the subject about when the circuit was working, and in another failure there were problems with the system software. A hardware failure caused termination of the final dialogue.

Table 2
Utterance breakdown into major subdialogues.

Subdialogue Type	Directive Mode		Declarative Mode	
	Average	Percent	Average	Percent
Introduction	2.9	5.2%	2.6	11.6%
Assessment	15.4	27.4%	7.9	35.1%
Diagnosis	11.8	21.0%	6.7	29.8%
Repair	2.9	5.2%	0.3	1.3%
Test	23.2	41.2%	5.0	22.2%
Total	56.2		22.5	

it is working, and the basic nature of repairs, but will need some assistance with diagnosing specific problems. Consequently, we would expect the following differences between modes for users who are able to take the initiative:

- Introduction Subdialogue: The number of utterances will change little, since problem introduction seems independent of initiative.
- Assessment Subdialogue: The number of utterances will be reduced slightly in declarative mode, as users who take the initiative may exploit their control of the dialogue to carry out some preliminary steps without verbal interaction.
- Diagnosis Subdialogue: The number of utterances will change little, since all users presumably need the computer's assistance in problem diagnosis.
- Repair Subdialogue: The change should be dependent on the task domain. If the repair process is basically the same once the error is diagnosed, few utterances will be required as repairs can be done without discussion. If the repair process is highly dependent on the type of error (e.g., debugging a program), even the skilled user may require significant advice from the system. For our domain, we expect a reduction in the number of utterances spoken in declarative mode, since the repair process (adding a wire) is similar across the different problem types.
- Test Subdialogue: The number of utterances is significantly reduced (i.e., users who take the initiative can verify the circuit behavior without dialogue).

6.2.2 Overall Averages. Table 2 shows the average and relative number of utterances spoken per dialogue in each of the main task subdialogues. The reported data combine both computer and user utterances. Note that virtually no utterances were ever spoken during the Repair phase of declarative mode dialogues. This is because the repair process was always the addition of a missing wire to the circuit, a process that users quickly became able to do without explicit guidance. However, since not many utterances were spoken in the Repair phase of the directive mode dialogues either, the major source of the reduction in the absolute number of utterances spoken per dialogue occurred in the Assessment, Diagnosis, and Test phases, especially the Test

Table 3
Statistical results on utterance classification for the first five problems.

	Mode Effect		Subdialogue Effect		Interaction Effect	
	F Value	*p*	*F* Value	*p*	*F* Value	*p*
By-subjects	$F(1,7) = 24.93$	$= 0.002$	$F(3,21) = 17.77$	< 0.001	$F(3,21) = 4.93$	$= 0.01$
By-items	$F(1,4) = 32.26$	$= 0.005$	$F(3,12) = 13.99$	< 0.001	$F(3,12) = 9.70$	$= 0.002$

phase. Although we originally expected little change in the number of utterances as a function of initiative for the Diagnosis phase, the large increase in the number of utterances spoken for that phase for problem 6, during directive mode interactions had a major impact on the overall averages. Excluding problem 6, the average number of utterances spoken in the Diagnosis phase was 9.4 in directive mode and 7.2 in declarative mode.

6.2.3 Statistical Analysis. While the first eight problems in each of the two experimental sessions are balanced (Section 4.2), we must distinguish between the first five problems of each session, where there was a single missing wire in each problem and problems 6 through 8 in each session, which have two missing wires. Not all subjects completed the same number of dialogues for problems 6 through 8 in the two experimental sessions. Consequently, including them in the computation of the average number of utterances spoken in a given subdialogue phase would distort the averages used in a statistical analysis.[10]

Therefore, we apply the statistical technique of analysis of variance (ANOVA) to the data from the first five problems of each session, the single-missing-wire problems. This represents a total of 60 completed dialogues. A 2 X 4 design (mode X subdialogue phase) was used (the Introduction phase was omitted).[11]

Table 3 summarizes the results of the statistical analysis. The analysis was conducted using the averages by subjects as well as by items (problems). The individual main effects showed very strong statistical significance under both forms of analysis while the interaction effect of mode and subdialogue phase also appears to be statistically significant, but not quite as strongly as the main effects individually. We now turn our attention to the order effect. Did the order in which subjects were given the initiative affect their performance?

6.2.4 The Effects of Experience. As mentioned in Section 4.2, we balanced the experiment problems according to type, such that problem *k* of both sessions 2 and 3 was the same type of problem. Furthermore, we balanced the subjects also. Half the subjects used the system when it was operating in directive mode for session 2 while the other half used the system when it was operating in declarative mode for session 2. The mode was, of course, reversed for session 3 for both groups. One of our claims has been that as users gain experience and are given the initiative by the system, they

10 As shown in the transition model of Figure 5, the Diagnosis, Repair, and Test subdialogue phases could occur twice in a dialogue with two missing wires.

11 As discussed in the next section, the order in which subjects were given the initiative did not show a significant effect and is omitted from the current analysis.

will take advantage of that. We might expect then, that subjects given the initiative in session 3 would behave differently than subjects given the initiative in session 2. Furthermore, we might expect difficulties for subjects given the initiative in session 2 who then had to work with the system in directive mode in session 3. What do we find in the results?

We conducted a paired t-test on the paired differences[12] in the average number of utterances spoken per dialogue between the two modes, as a function of the problem number. Computing this test statistic for the two subdialogue phases in the domain where we would expect additional experience to have the most effect, Assessment and Diagnosis, yields the following results. For the Assessment phase, the test statistic is 0.854 with a corresponding p value of 0.42 for 7 degrees of freedom. For the Diagnosis phase, the test statistic is 0.556 with a corresponding p value of 0.60. Consequently, we do not find that the order in which a subject was given the initiative has a significant effect on the number of utterances spoken in a given subdialogue phase. We do not find this result surprising because:

- Some expertise was gained during the preliminary training session, so some subjects were ready to be given initiative in session 2. In fact, the two subjects who struggled with using declarative mode in session 2 only contribute 5 of the 48 declarative mode data points used in computing the averages.
- Some subjects, as part of their expertise, developed a somewhat ritualistic style of interaction with the machine, which may have lengthened their interactions.

6.3 User Initiation of Subdialogue Transitions

When the computer has total control of the dialogue, in directive mode, it is expected that the computer will initiate the transitions between subdialogues. How will this change when the computer operates in declarative mode and control is given back to the user?

While user control means the user's goals have priority, it does not necessarily mean the user will initiate every transition from one subdialogue to the next. The user controls the dialogue but still requires computer assistance. Consequently, it is expected that the computer will still initiate many of the transitions to the Assessment and Diagnosis phases in order to provide assistance in these areas, but that the user will be able to transition to other subdialogues as deemed appropriate. In particular, it is expected that the user will initiate most of the transitions to the final Test phase for confirming circuit behavior, since an experienced user would have learned how the circuit should function.

These hypotheses are generally supported by the results in Table 4. When the computer had the initiative (the directive mode dialogues), very few subdialogue transitions were ever initiated by the user other than to the final Test phase when the repair would cause the circuit to begin to function normally. When the computer yielded the initiative (the declarative mode dialogues), users initiated the transition

12 For example, the value of 12 for problem 3 in the Assessment phase for subjects who operated in declarative mode in session 2 and directive mode in session 3 is obtained by subtracting the declarative mode average for the number of Assessment utterances spoken per dialogue, 9, from the directive mode average, 21. This value would be paired with the value 8 (18 – 10) also for problem 3 in the Assessment phase, but for subjects who operated in directive mode in session 2 and declarative mode in session 3.

Table 4
Subdialogues initiated by each participant.

Assessment Subdialogues	Directive Mode	Declarative Mode
System-initiated	70	83
User-initiated	9	20
Diagnosis Subdialogues	**Directive Mode**	**Declarative Mode**
System-initiated	96	96
User-initiated	0	7
Repair Subdialogues	**Directive Mode**	**Declarative Mode**
System-initiated	78	3
User-initiated	0	5
Test Subdialogues	**Directive Mode**	**Declarative Mode**
System-initiated	71	6
User-initiated	22	77

to the final stage of the dialogue almost every time. In the intermediate stages, the computer still initiated most subdialogues, but users occasionally felt compelled to cause a change to a different phase. This rarely happened when the computer had the initiative. Not counting the Introduction, which had to be initiated by the computer, only 9% of all subdialogues in directive mode were initiated by the user while 37% of the subdialogues in declarative mode were user-initiated.

6.4 General Subdialogue Transitions

As described in Section 5.2, the natural course of transition from subdialogue to subdialogue is described by the following regular expression:

$$I^+A^+(D^+R^*T^+)^nF$$

where n represents the number of individual repairs in the problem (i.e., number of missing wires in our domain). If every dialogue followed this model, then we would expect to see all transitions out of the Introduction phase go to the Assessment phase, all transitions out of the Assessment phase go to the Diagnosis phase, and all transitions out of the Repair phase go to the Test phase. However, with the potential for miscommunication as well as the potential for users to exploit their expertise and control of the dialogue to skip discussion of some task steps, it is highly unlikely that the actual results will follow the idealized model. Where might we see differences?

Table 5 shows the actual breakdown in percentages. The row value represents the initial subdialogue phase and the column represents the new subdialogue. The *F* column represents the finished state (i.e., dialogue completion). For example, the percentage of all transitions out of the Diagnosis phase that went to the Assessment phase is 18.8% in directive mode and 38.8% in declarative mode. The *X* entries along the main diagonal represent impossible exit transitions (i.e., there cannot be a transition from Diagnosis to Diagnosis). The "—" entries represent values of less than 5%.[13] If the dialogues follow the transition model, then the largest entries should be in the values

13 Consequently, the numerical values in each row will not necessarily add up to 100%.

Table 5
Subdialogue transition breakdown as a function of dialogue mode.

	Directive Mode						Declarative Mode						
	I	A	D	R	T	F		I	A	D	R	T	F
I	X	**100**	—	—	—	—	I	X	**96.8**	—	—	—	—
A	—	X	**91.1**	—	7.6	—	A	—	X	**79.6**	—	19.4	—
D	—	18.8	X	**68.7**	12.5	—	D	—	38.8	X	7.8	**53.4**	—
R	—	—	—	X	**96.2**	—	R	—	12.5	12.5	X	**75.0**	—
T	—	—	24.7	12.9	X	**62.4**	T	—	—	24.1	—	X	**72.3**

in the diagonal just above the main diagonal. The resulting largest entry in each row is noted in boldface.

For the most part, the percentages are consistent with the model, especially in the early phase transitions and in the transitions out of the Test subdialogue. Based on the relative number of completed dialogues that required the repair of two missing wires (17 in directive mode, 21 in declarative mode), the expected percentage of transitions from Test-to-Diagnosis would be 22.7% in directive mode and 25.9% in declarative mode.[14] The actual values of 24.7% and 24.1% compare favorably with the expected results. The large relative difference in percentages for transitions from Diagnosis to either Repair or Test in the two modes is also expected, given that users who take the initiative can make the repair themselves without discussing it with the computer. The transition percentages that are most surprising are the Diagnosis-to-Assessment transitions in both modes and the Test-to-Repair transitions in directive mode. The Diagnosis-to-Assessment transitions are indicative of attempts at error correction. That is, at some point during Diagnosis either the computer or the user becomes suspicious of the initial problem assessment and consequently moves back to Assessment to be sure that the erroneous circuit behavior is properly understood. The Test-to-Repair transition is common when the user makes the repair without mentioning it. That is, the user has prematurely moved from Repair to Test without notifying the computer that the repair has actually been made. In directive mode dialogues, the computer will require verbal verification of the repair before transitioning to the Test phase.

In general, 64% of the dialogues in directive mode have no "unusual" transitions (where we define unusual as a transition not described by our model). In contrast, only 33% of the declarative mode dialogues had no unusual transitions, again demonstrating how users felt free to skip steps without discussion. This particularly increased as users gained more experience, with only 26% of the 35 declarative dialogues of the final session containing no unusual transitions.

6.5 Task Control versus Linguistic Control

As described in Section 2.1, our view of initiative concerns which participant's task goals currently have priority. Walker and Whittaker's (1990) study of mixed-initiative dialogue used a notion of control based on linguistic goals as specified in the control rules first presented in Section 2.3 and repeated below. These rules are a function of

14 These Test-to-Diagnosis transitions occur because after repairing one of the missing wires, the Test phase would show that the circuit is still not working due to the other missing wire, causing a transition back to the Diagnosis phase to discover the other problem.

the classification of the linguistic goal of the current utterance (Assertion, Command, Question, or Prompt) and reflect the status of initiative after the utterance was made.

1. Assertion: The speaker has the initiative unless the utterance is a response to a Question.
2. Command: The speaker has the initiative.
3. Question: The speaker has the initiative unless the utterance is a response to a question or command.
4. Prompt: The hearer has the initiative.

We analyzed our dialogues using this notion of control with one modification—assertions that were a continuation of the current topic left the initiative unchanged. Consider the following dialogue excerpt:

```
C: The LED is supposed to be displaying an alternately flashing
     one and seven.
U: The LED is off.
C: The power is on when the switch is up.
U: The switch is up.
C: The switch is connecting to the battery when there is a wire
     between connectors 111 and 120.
```

In both cases the user's assertions continue the topic introduced by the computer and do not cause a change of control. Contrast this with the following:

```
C: The LED is supposed to be displaying an alternately flashing
     one and seven.
U: There is no wire between connector eight four and connector
     nine nine.
C: There is supposed to be a wire between connector 84 and
     connector 99.
```

In this case the user's assertion does change control, as it is a change of topic. Our rule modification reflects this issue.

6.5.1 Hypotheses. The two primary measures reported by Walker and Whittaker are average number of utterances between control shifts and percent of total utterances controlled by the computer. Their results for task-oriented dialogues about constructing a water pump showed that experts had control of the dialogue about 90% of the time. In contrast, their results for advisory dialogues where clients talked to an expert over the phone to obtain assistance in diagnosing and repairing various software faults showed that experts had control only about 50% of the time. While our problem domain is more similar to the advisory dialogues, the nature of our dialogues is more similar to the task-oriented dialogues as the task of circuit repair is being completed concurrently with the dialogue. Therefore, we expect the computer to show strong linguistic control when it has task initiative. Conversely, when users control the task initiative, we expect more assertions by the user concerning the user's own task goals, rather than direct responses to computer questions or commands. Nevertheless, because the computer is the ultimate expert, we still expect it to respond with assertions of facts designed to assist the user that take a linguistic form that would be classified as continuing or regaining linguistic control (e.g., "The power is on when the switch

Table 6
Differences in linguistic control as a function of initiative.

	Directive Mode	Declarative Mode
Percentage of total utterances controlled by the computer	97.6	85.7
Average number of utterances between control shifts	15.8	3.3

Table 7
Differences in average number of utterances between control shifts.

Problem	Number of Paired Differences	Mean	Standard Deviation
1	5	21.2	24.5
2	5	17.1	10.8
3	4	39.8	13.0
4	4	31.4	12.2
5	4	16.7	19.5
6	7	23.3	22.6
7	5	16.2	6.9
8	4	31.7	13.0

is up," from the first excerpt). The net effect should be that user task control in declarative mode will lead to more frequent linguistic control shifts although the computer will still have overall control of most utterances.

6.5.2 Resulting Comparison. Table 6 gives the results. Although the user has linguistic control only 14.3% of the time in declarative mode, this is much more often than in directive mode. Correspondingly, the average number of utterances between control shifts is reduced by a factor of almost 4.8. A detailed examination shows that 79% of the 248 control shifts were caused either by the user attempting to correct a computer misunderstanding (Section 6.6.2) or by the user initiating a task topic change by asserting new task information. These types of control shifts occurred once every 4.4 user-utterances in declarative mode, but only once every 32.0 user-utterances in directive mode. The remaining control shifts were due to requests for repetition of the previous utterance or requests for other information. Table 7 presents the mean difference in the average number of utterances between control shifts for each of the balanced problems. Thus, the value 21.2 for problem 1 means that the difference in the average number of utterances between control shifts was greater by 21.2 utterances in directive mode over declarative mode. These results show that there is a relationship between our notion of task control and the Whittaker and Stenton (1988) notion of linguistic control evaluated by Walker and Whittaker (1990)—namely, that as users exploit their task expertise, linguistic control shifts occur much more frequently. This result may prove useful as a possible cue for when the system needs to release task initiative to the user during a mixed-initiative dialogue—as linguistic control shifts begin to occur more frequently, it may be an indicator that a user is gaining experience and can take more overall control of the dialogue. Further development and testing of this hypothesis are needed.

6.6 The Impact of Miscommunication

One important phenomenon of interactive dialogue that has recently begun to receive attention in the computational linguistics community is the handling of miscommunication (e.g., McRoy and Hirst [1995], Brennan and Hulteen [1995], and Lambert and Carberry [1992]). In the Circuit Fix-It Shop the computer misunderstood user-utterances 18.5% of the time. The primary cause of these misunderstandings was the misrecognition of the words spoken by the user—only 50% of the user's utterances were correctly recognized word for word. Consequently, misunderstanding occurred more often in declarative mode (24.7% of user-utterances) than directive mode (15.0% of user-utterances). This is due to the fact that, on average, users spoke longer utterances in declarative mode. Speech recognition technology has improved dramatically since this system was tested, but the need for handling miscommunication is still relevant as users and designers will continually test the performance limits of available technology. Human-human communication frequently contains miscommunication, so we should expect it in human-computer dialogue as well. For the current system, how did miscommunication impact on the dialogue structure?

6.6.1 Frequency of Experimenter Interaction. As mentioned in Section 4.4, when the computer made a serious misinterpretation the experimenter was allowed to tell the user about the computer's erroneous interpretation without telling the user what to do. Computer misinterpretation of the user's utterances due to misrecognition of words can cause confusion between the user and computer, and ultimately, failure of the dialogue. With the computer running in declarative mode, the experimenter chose to make such statements once every 8.5 user-utterances, but only once every 26.5 user-utterances in directive mode. Not all misrecognitions required experimenter interaction.[15]

6.6.2 User-Initiated Corrections. The previously mentioned procedure for notifying the user of a serious misrecognition leaves the responsibility with the user to try to correct the computer's misunderstanding. It is hypothesized that when the computer has yielded the initiative, users are more likely to attempt to redirect the computer's focus when an error situation occurs. Conversely, users will tend to give up trying to redirect the computer's attention when the computer has the initiative because the machine will proceed on its own line of reasoning, ignoring what it perceives as user interrupts even when these interrupts are actually attempts at resolving previous miscommunications. This is borne out by the results. Overall, while the computer was operating in directive mode, the user attempted to correct only 24% of the misunderstandings for which the user received notification. In contrast, while the computer was operating in declarative mode, the user attempted to correct 52% of the misunderstandings.

15 As reported in Smith and Gordon (1996), there were a total of 250 misunderstandings in declarative mode, 215 for which the experimenter was allowed to notify the user. The experimenter chose to intervene in 118 of these or 54% of the time. In contrast, there were a total of 276 misunderstandings in directive mode, 226 for which the experimenter was allowed to notify the user. In only 69 or 30.5% of these misunderstandings did the experimenter notify the user. The difference in the relative number of notifications is largely due to the fact that, in directive mode, the computer frequently ignored the statements it misunderstood, as the misunderstandings often were in conflict with the computer's current task goal. Consequently, it was unnecessary for the experimenter to notify the user about such misunderstandings since they would not cause a problem. On the other hand, confusion between computer and user was much more likely in declarative mode because the computer would more frequently formulate a response based on its erroneous interpretation of the user's input. In these cases, there was a greater need for the experimenter to notify the user of the misunderstanding.

6.7 Summary of Results

What general conclusions can we draw from this analysis? Based on the evaluation of the Circuit Fix-It Shop at two different levels of initiative, we have observed the following phenomena:

- Directive mode dialogues tend to follow an orderly pattern consisting largely of computer-initiated subdialogue transitions, terse user responses, and predictable subdialogue transitions. However, the inflexibility of this mode is a severe drawback in the presence of user-correctable miscommunications.
- Declarative mode dialogues are shorter but less orderly, consisting of more user-initiated subdialogue transitions. There is evidence that users are willing to modify their behavior as they gain expertise, provided the computer allows it. The ability to yield the initiative as users gain experience is essential if a dialogue system is to be useful in practical applications involving repeat users.
- The small number of subjects and the design of the experiment make it difficult to observe differences within a given level of initiative as subjects gain additional expertise. Nevertheless, in a practical environment we believe the capacity to change initiative during a dialogue is essential for obtaining the most effective interaction between repeat users and a system. It is our conjecture that being able to vary initiative between dialogues is insufficient, but further study of this issue is needed.

After reviewing other empirical studies in the next section, we will address the impact of these results on future research in Section 8.

7. Recent Empirical Studies Relevant to Human-Computer Mixed-Initiative Dialogue Structure

Danieli and Gerbino (1995) also look at dialogues with an implemented computer system. This system answers user queries about train schedules and services. The focus of the paper is on a few objective and several subjective performance measures of two interaction strategies similar to the directive and declarative modes described in this paper. Their paper concludes that the mode similar to our directive mode is more robust and more likely to succeed, but the mode similar to our declarative mode is faster and less frustrating to experienced users. The general performance results obtained during our testing of the Circuit Fix-It Shop (Section 6.1) lend support to their claim, as 88% of our attempted dialogues in directive mode were completed successfully, compared to 80% in declarative mode, and experimenter interaction (of any kind) occurred only once every 18 user-utterances in directive mode, but once every 6 user-utterances in declarative mode. While their dialogue control algorithms are not identical to ours, their results are complementary, as they show that performance differences as a function of the computer's level of control may be prevalent in database query interactions as well.

Guinn (1996) reports on the utility of computer-computer dialogue simulations of the Collaborative Algorithm, an extension of our Missing Axiom Theory (Section 3.1) for modeling dialogue processing. Guinn has implemented the model and run extensive simulations of computer-computer dialogues in order to explore the dynamic

setting of initiative as the dialogue ensues. The model attaches an initiative level to each task goal, and a competency evaluation, based on user model information, is used to decide who should be given the initiative for a given task goal. There is ongoing work in implementing and testing the Collaborative Algorithm in human-computer interactive environments.

8. Conclusions

While there is ample analysis of dialogue structure based on human-human and simulated human-computer dialogue, there is very little information on the structure of actual human-computer dialogue. In this paper we have reviewed an integrated approach to dialogue processing that allows a system to support variable initiative behavior in its interactions with human users. Furthermore, we have reported the results from the analysis of 141 dialogues collected during experimental use with a system based on the overall dialogue-processing model. These results indicate differences in both dialogue structure and user behavior as a function of the computer's level of initiative. An important open question is the degree to which the parameters of the transition model for task-oriented dialogues for repair assistance are domain dependent. For example, the relative amount of time spent in each subdialogue phase is likely to be highly dependent on the domain. Furthermore, the model does not fully take into account different types of miscommunication and their repair. These issues require further study.

The next step in extending the dialogue-processing model is to incorporate the knowledge gained from this study in addressing two of the most significant unresolved problems in human-computer dialogue: (1) automatic switching of initiative during dialogue; and (2) automatic detection and repair of miscommunication. The current dialogue-processing model considers subdialogues at the lower level of basic domain actions. Extending the model to describe dialogue structure at the more abstract level of task phases would allow the system to track the excessive and unusual subdialogue transitions observed in this study. Such tracking can be used for recognizing evolving user expertise as well as detecting a lack of mutual understanding about the current situation. Normally, implemented dialogue systems tend to be based on processing models that are rich in domain information, but are deficient in one or more areas of knowledge about dialogue. Incorporating more metaknowledge about dialogue structure into the model should lead to more human-like performance in handling initiative changes and miscommunication problems.

The observations reported in this paper are an initial step on the long road to a comprehensive model of actual human-computer dialogue structure. It is hoped that these results will encourage other researchers to construct experimental NL dialogue systems, test these systems, and then analyze and report the results so that a more comprehensive view of human-computer dialogue structure can be obtained. In general, we believe that the natural life cycle of experimental NL dialogue systems should be one of analyzing, modeling, building, and testing so that the analysis of actual human-computer dialogues can lead to the development of more effective systems. Such a methodology allows us to gain clearer insight into the evolving nature of human-computer dialogues.

Acknowledgments

The authors wish to express their thanks to Robert M. Hoekstra, Lynn Eudey, and the anonymous reviewers who advised us concerning the appropriateness of various statistical tests and data displays. We are further grateful to the anonymous reviewers of *Computational Linguistics* for their helpful

comments about previous drafts of this paper. A special note of thanks goes to guest editor Marilyn Walker, who provided numerous constructive suggestions about the form of the introductory material. Other researchers who contributed to the development of the experimental system include Alan W. Biermann, Robert D. Rodman, Ruth S. Day, D. Richard Hipp, Dania Egedi, and Robin Gambill. The writing of this paper has been supported by National Science Foundation Grant NSF-IRI-9501571.

References

Allen, James F., Alan M. Frisch, and Diane J. Litman. 1982. ARGOT: The Rochester dialogue system. In *Proceedings of the 2nd National Conference on Artificial Intelligence*, pages 66–70.

Allen, James F., Lenhart K. Schubert, George Ferguson, Peter Heeman, Chung Hee Hwang, Tsuneaki Kato, Marc Light, Nathaniel Martin, Bradford Miller, Massimo Poesio, and David R. Traum. 1995. The TRAINS project: A case study in building a conversational planning agent. *Journal of Experimental and Theoretical Artificial Intelligence*, 7:7–48.

Biermann, Alan W., Linda Fineman, and J. Francis Heidlage. 1992. A voice- and touch-driven natural language editor and its performance. *International Journal of Man-Machine Studies*, 37:1–21.

Bobrow, D. G., R. M. Kaplan, M. Kay, D. A. Norman, H. Thompson, and T. Winograd. 1977. GUS, a frame driven dialog system. *Artificial Intelligence*, 8:155–173.

Brennan, Susan E. and Eric A. Hulteen. 1995. Interaction and feedback in a spoken language system: A theoretical framework. *Knowledge-Based Systems*, 8:143–151.

Carberry, Sandra. 1988. Modeling the user's plans and goals. *Computational Linguistics*, 14(3):23–37.

Dahlbäck, Nils, Arne Jönsson, and Lars Ahrenberg. 1993. Wizard of Oz studies—why and how. *Knowledge-Based Systems*, 6(4):258–266.

Danieli, Morena and Elisabetta Gerbino. 1995. Metrics for evaluating dialogue strategies in a spoken language system. In *Proceedings of the 1995 AAAI Spring Symposium on Empirical Methods in Discourse Interpretation and Generation*, pages 34–39.

Fraser, Norman M. and G. Nigel Gilbert. 1991. Simulating speech systems. *Computer Speech and Language*, 5:81–99.

Frederking, Robert E. 1988. *Integrated Natural Language Dialogue: A Computational Model.* Kluwer Academic Publishers, Boston.

Grosz, Barbara J. 1978. Discourse analysis. In D. E. Walker, editor, *Understanding Spoken Language*. North-Holland, New York, pages 235–268.

Grosz, Barbara J. and Candace L. Sidner. 1986. Attentions, intentions, and the structure of discourse. *Computational Linguistics*, 12(3):175–204.

Guinn, Curry I. 1996. Mechanisms for mixed-initiative human-computer collaborative discourse. In *Proceedings of the 34th Annual Meeting*, pages 278–285. Association for Computational Linguistics.

Hafner, Carole D. and Kurt Godden. 1985. Portability of syntax and semantics in Datalog. *ACM Transactions on Office Information Systems*, pages 141–164, April.

Heeman, Peter and James Allen. 1994. Detecting and correcting speech repairs. In *Proceedings of the 32nd Annual Meeting*, pages 295–302. Association for Computational Linguistics.

Hendrix, Gary G., Earl D. Sacerdoti, Daniel Sagalowicz, and Jonathan Slocum. 1978. Developing a natural language interface to complex data. *ACM Transactions on Database Systems*, pages 105–147, June.

Hoeppner, W., T. Christaller, H. Marburger, K. Morik, B. Nebel, M. O'Leary, and W. Wahlster. 1983. Beyond domain-independence: Experience with the development of a German language access system to highly diverse background systems. In *Proceedings of the 8th International Joint Conference on Artificial Intelligence*, pages 588–594.

Isard, Amy and Jean Carletta. 1995. Replicability of transaction and action coding in the map task corpus. In *Proceedings of the 1995 AAAI Spring Symposium on Empirical Methods in Discourse Interpretation and Generation*, pages 60–66.

Jullien, C. and J. Marty. 1989. Plan revision in person-machine dialogue. In *Proceedings of the Fourth Conference of the European Chapter of the Association for Computational Linguistics*, pages 153–160.

Kaplan, S. J. 1982. Cooperative responses from a portable natural language query system. *Artificial Intelligence*, 19(2):165–187.

Kitano, Hiroaki and Carol Van Ess-Dykema. 1991. Toward a plan-based understanding model for mixed-initiative dialogues. In *Proceedings of the 29th Annual Meeting*, pages 25–32. Association for

Computational Linguistics.
Lambert, Lynn and Sandra Carberry. 1992. Modeling negotiation subdialogues. In *Proceedings of the 30th Annual Meeting*, pages 193–200. Association for Computational Linguistics.
Levine, J. M. 1990. PRAGMA—A flexible bidirectional dialogue system. In *Proceedings of the 8th National Conference on Artificial Intelligence*, pages 964–969.
Litman, Diane J. and James F. Allen. 1987. A plan recognition model for subdialogues in conversations. *Cognitive Science*, 11(2):163–200.
McRoy, Susan and Graeme Hirst. 1995. The repair of speech act misunderstandings by abductive inference. *Computational Linguistics*, 21(4):435–478.
Moody, Terry S. 1988. *The Effects of Restricted Vocabulary Size on Voice Interactive Discourse Structure*. Ph.D. thesis, North Carolina State University.
Oviatt, Sharon. 1995. Predicting spoken disfluencies during human-computer interaction. *Computer Speech and Language*, 9:19–35.
Oviatt, Sharon L. and Philip R. Cohen. 1991. Discourse structure and performance efficiency in interactive and non-interactive spoken modalities. *Computer Speech and Language*, 5:297–326.
Peckham, Jeremy. 1991. Speech understanding and dialogue over the telephone: An overview of progress in the SUNDIAL project. In *Proceedings of the 2nd European Conference on Speech Communication and Technology*, pages 1469–1472.
Seneff, Stephanie. 1992. TINA: A natural language system for spoken language applications. *Computational Linguistics*, 8(1):61–86.
Smith, Ronnie W. 1991. *A Computational Model of Expectation-Driven Mixed-Initiative Dialog Processing*. Ph.D. thesis, Duke University.
Smith, Ronnie W. 1992. Integration of domain problem solving with natural language dialog: The missing axiom theory. In *Proceedings of Applications of AI X: Knowledge-Based Systems*, pages 270–278.
Smith, Ronnie W. and Steven A. Gordon. 1996. Pragmatic issues in handling miscommunication: Observations of a spoken natural language dialog system. In *Proceedings of the AAAI '96 Workshop on Detecting, Repairing, and Preventing Human-Machine Miscommunication*, pages 21–28.
Smith, Ronnie W. and D. Richard Hipp. 1994. *Spoken Natural Language Dialog Systems: A Practical Approach*. Oxford University Press, New York.
Smith, Ronnie W., D. Richard Hipp, and Alan W. Biermann. 1995. An architecture for voice dialog systems based on Prolog-style theorem-proving. *Computational Linguistics*, 21(3):281–320.
Walker, Marilyn and Steve Whittaker. 1990. Mixed initiative in dialogue: An investigation into discourse segmentation. In *Proceedings of the 28th Annual Meeting*, pages 70–78. Association for Computational Linguistics.
Waltz, David L. 1978. An English language question answering system for a large relational database. *Communications of the ACM*, pages 526–539, July.
Whittaker, Steve and Phil Stenton. 1988. Cues and control in expert-client dialogues. In *Proceedings of the 26th Annual Meeting*, pages 123–130. Association for Computational Linguistics.
Whittaker, Steve and Phil Stenton. 1989. User studies and the design of natural language systems. In *Proceedings of the Fourth Conference of the European Chapter of the Association for Computational Linguistics*, pages 116–123.
Wilensky, Robert, David N. Chin, Marc Luria, James Martin, James Mayfield, and Dekai Wu. 1988. The Berkeley UNIX consultant project. *Computational Linguistics*, 14(4):35–84.
Young, Sheryl R., Alexander G. Hauptmann, Wayne H. Ward, Edward T. Smith, and Philip Werner. 1989. High level knowledge sources in usable speech recognition systems. *Communications of the ACM*, pages 183–194, February.
Young, S. J. and C. E. Proctor. 1989. The design and implementation of dialogue control in voice operated database inquiry systems. *Computer Speech and Language*, 3:329–353.