INITIATIVE-DEPENDENT FEATURES OF HUMAN-COMPUTER DIALOGS IN A TASK-ASSISTANCE DOMAIN

Ronnie W. Smith
Department of Mathematics
Computer Science Subarea
East Carolina University
Greenville, NC 27858, USA
rws@math1.math.ecu.edu

Abstract

This paper presents information about the dialog structure of actual human-computer dialog obtained from the analysis of 141 dialogs collected during experimental interaction with a variable initiative spoken natural language dialog system organized around the paradigm of the Missing Axiom Theory for language use. The reported results indicate there are differences in user behavior and dialog structure as a function of the computer's level of initiative. These results provide evidence that a practical spoken natural language dialog 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 DIALOG

It is generally acknowledged that developing a successful computational model of interactive natural language (NL) dialog requires extensive analysis of sample dialogs. Much analysis of human-human interactions has been done. Important recent work includes Walker and Whittaker, 1990 on mixed initiative in dialog, and Oviatt and Cohen, 1991 on the comparison of interactive and non-interactive spoken modalities. However, there are limitations to this analysis. As noted by Fraser and Gilbert, 1991, "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." Further support for the study of actual human-computer dialog is provided in Oviatt and Cohen, 1991: "...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 communication ..." 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."

Fortunately, at least one system, The Circuit Fix-It Shop, has been built and tested in 141 dialogs totaling 2840 user utterances Smith and Hipp, 1994. This system was constructed to test the effectiveness of a specific theory for providing an integrated model of dialog processing, the Missing Axiom Theory. This paper first overviews how the theory facilitates integrated dialog processing and then presents some analytical results on the differences in human-computer dialog structure according to the computer's level of initiative.

2. DIALOG PROCESSING MODEL: AN INTEGRATED APPROACH

Most prior work on natural language dialog has either focused on individual subproblems such as quantification, presuppositions, ellipsis, anaphoric reference, plan recognition, and user modeling, or else focused on dialog processing issues in database query applications. However, there has been little work on integrating the various aspects of dialog processing into a unified whole (exceptions are Allen et al., 1995 and Young et al., 1989). Consequently, a dialog processing model was developed (Smith et al., 1995 and Smith and Hipp, 1994) that when implemented in an electronic repair domain exhibits a number of important behaviors including: (1) problem-solving; (2) coherent subdialog movement; (3) user model usage; (4) expectation usage; and (5) variable initiative behavior.

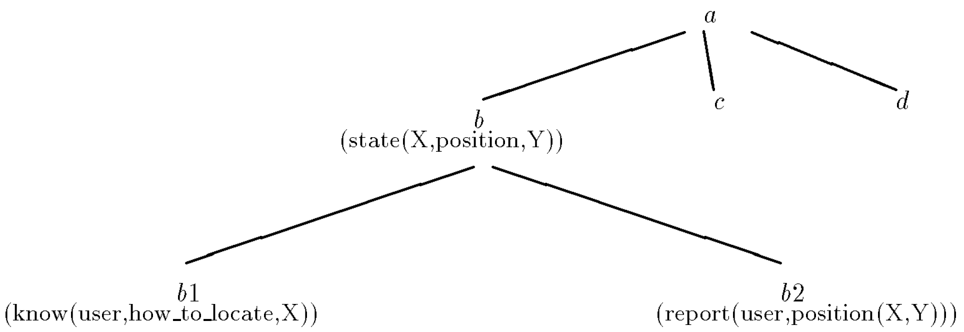

Figure 1: Sample theorem description

The key aspect of the model that enables this integration is the Missing Axiom Theory Smith, 1992 for driving the dialog process. Integrating the necessary domain reasoning with a general NL dialog mechanism requires a reasoning system for achieving goals. Theorem proving, a well understood computational approach, was selected for the reasoning mechanism. The theory then utilizes language interaction to acquire the missing axioms needed for proving a target goal. Consider the proof tree of figure 1. a is proven by proving b, c, and d; and b is proven by proving $b1$ and $b2$. Suppose that the process of proving a yields subgoal b, `state(X,position,y` (i.e. that the position of object X is Y), which leads to subgoals $b1$ (`know(user,how_to_locate,X)`) and $b2$ (`report(user,position(X,Y))`). Since $b1$ and $b2$ are problem-dependent facts not ordinarily available to the machine, the system resorts to natural language in this case and interacts with the user to status. $b1$ and $b2$ are known as missing axioms and could yield the dialog shown below (C denotes the computer and U the user).

```
1.  C: The knob is at the top of the green region.
2.  U: I see it.
3.  C: What is its setting?
4.  U: Zero.
```

Completion of this dialog enables the proof of b and possibly, at a later time a. The example also illustrates how user modeling is done. For example, if an earlier interaction had involved successfully referencing the knob, then $b1$ would be included as an achieved fact in the user model. The proof of b would have achieved $b1$ trivially and omitted the first comment about locating the knob (utterance 1).

Furthermore, a close correlation between subdialogs and domain actions is obtained. A subdialog consists of all the language interaction pertaining to a domain action. Movement between and within subdialogs is based on determining the domain action to which a user input is relevant. This requires using *expectations* about the user's next utterance based on the current set of active goals. For example, if the computer said, “Add a wire between connector 16 and 29,” possible expected user responses include (1) that the wire was added; (2) that the wire exists; (3) that the user does not know how to add the wire; (4) that the user needs help adding the wire; and (5) that the user does not know how to identify the location of the wire.

The use of expectations has been previously studied by researchers interested in speech recognition (Young and Proctor, 1989 and Young et al., 1989) and plan recognition (Carberry, 1990). By searching through different sets of expected responses, the computer can determine which goal is relevant to a user's utterance. Movement to a goal other than the currently active one is termed an *interrupt* of the current subdialog. For example, if the user had responded with “The switch is stuck,” instead of “I see it,” for utterance 2 of the sample dialog, that would be an interrupt of the subdialog about subgoal $b1$ of figure 1. The way in which the computer responds to interrupts is a key factor in enabling variable initiative dialog to occur as discussed below.

3. VARIABLE INITIATIVE: GIVING PRIORITY TO A CONVERSANT'S GOALS

The need for variable initiative dialog arises because at some points the user has sufficient knowledge to take control and accomplish several goals without much computer assistance while at other times, the user

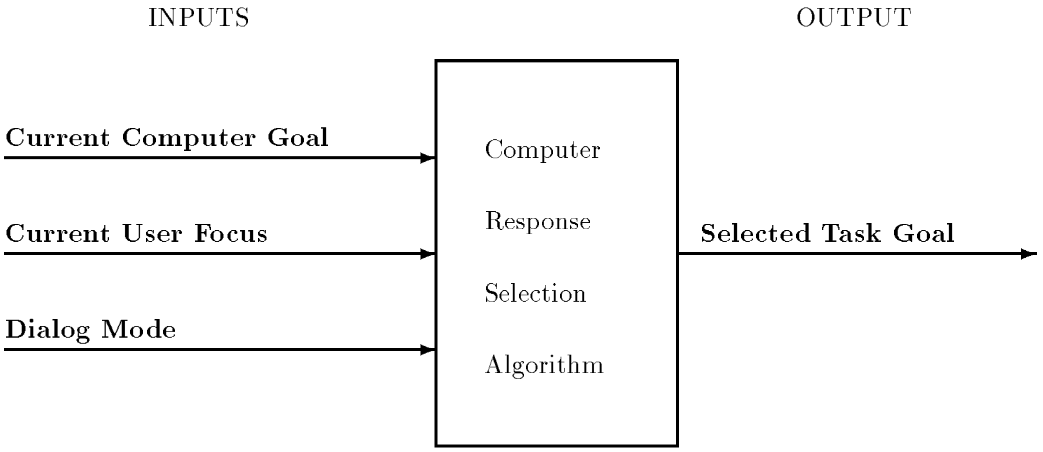

Figure 2: Flow diagram of computer response selection process

needs detailed assistance. Thus, computer initiative is characterized by giving priority to its own specific goals. However, initiative is not an all or nothing situation. Either the user or the computer may have the initiative without having complete control of the dialog. Based on this idea we have identified four dialog *modes* that characterize the level of initiative that the computer can have in the dialog. These are described below.

1. *directive*—The computer has complete dialog control, and allows no interruptions to other subdialogs. It recommends a task step and will use whatever dialog is necessary to obtain the needed item of knowledge related to the task step.
2. *suggestive*—The computer still has dialog control, but will allow minor interruptions to closely related subdialogs.
3. *declarative*—The user has dialog control and can interrupt to any desired subdialog, but the computer is free to mention relevant facts as a response to the user's statements.
4. *passive*—The user has complete dialog control; consequently, the computer will provide domain information only as a direct response to a user question.

The key observation is that the degree to which the computer allows user interrupts increases from directive to passive mode. Since the computer can indicate its level of initiative only through its responses, the choice of response topic is affected by the dialog mode. Details about the response process are provided in Smith and Hipp, 1994. Here we provide a brief overview. As shown in figure 2, the response process is a function of the computer's goal, the user focus, and the dialog mode. In an interrupt situation, the user focus differs from the computer's goal. In this case the dialog mode affects the selection process as follows.

1. Directive mode—The computer goal is selected without any regard for the user's focus.
2. Suggestive mode—If a common relationship can be found between the computer goal and user focus,[1] select this as the next goal; otherwise, select the computer's original goal.
3. Declarative mode—Search for a common relationship as in the suggestive mode case and select this fact; otherwise, select an uncommunicated fact relevant to the user focus.

[1]The search for a common relationship is a search through the domain knowledge hierarchy. For example, in our implemented domain of circuit repair, the circuit is decomposed into subcircuits where each subcircuit is decomposed into electronic components such as switches, wires, and lights. A common relationship is found when the user focus and computer goal share a sufficiently close common ancestor in the knowledge hierarchy.

4. Passive mode—Acknowledge that the computer has processed the user's last utterance.

Consider for example the situation where the user has stated, "the light is off," and suppose the computer knows that in order for the light to be lit, the switch must be turned up to activate the power. Consequently, the computer goal of highest priority is to put the switch up. However, any of the following goals could be selected, depending on the computer's level of initiative.

1. Directive mode—User focus is ignored with no interrupts permitted. The selected goal is to put the switch up.

2. Suggestive mode—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. Declarative mode—User now has control. Consequently, the selected goal must be a relevant fact. The previous goal is converted to "user learn that the power is on when the switch is up."

4. Passive mode—User has complete control. Computer must content itself to acknowledge processing the last user utterance.

This response selection process has been implemented as part of the previously mentioned Circuit Fix-It Shop. The two dialog segments shown below, obtained from actual usage of the implemented system, illustrate differences between mode for the two modes in which the system was experimentally evaluated: directive and declarative.

```
   Dialog Segment 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?

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

    Dialog Segment 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 the following phenomena from these dialog segments.

1. In the directive mode dialog segment the subject is performing task goals under the close guidance of the computer. There is language interaction about each task goal.

2. In the declarative mode dialog segment the subject independently carries out several task goals known to be necessary without any interaction. By allowing the user to arbitrarily change subdialogs, the computer is able to provide the relevant assistance when the potential problem is reported without requiring language interaction for task goals already completed.

Previous studies of variable initiative theory includeWalker and Whittaker, 1990, and Kitano and Van Ess-Dykema, 1991. In addition to a theory, we provide a mechanism that allows a computer to participate in such dialogs. After briefly describing some features of the system implementation, the remainder of this paper focuses on observed differences in the structure of the dialogs as a function of the computer's level of initiative.

4. IMPLEMENTATION RESULTS

The system was originally implemented and tested on a Sun 4 workstation with the majority of the code written in Quintus Prolog,[2] and the parser in C. The implemented dialog system assists users in repairing a Radio Shack 160 in One Electronic Project Kit. The particular circuit being used causes the Light-Emitting Diode (LED) to alternately display a one and seven. 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[3] DTCO1 text-to-speech converter is used to provide spoken output by the computer.

After testing system prototypes with a few volunteers, eight subjects used the system during the formal experimental phase. After a warmup session where the subject trained on the speech recognizer and practiced using the system, each subject participated in two sessions where up to ten problems were attempted. Subjects attempted a total of 141 dialogs of which 118 or 84% were completed successfully.[4] The system was able to find the correct meaning for 81.5% of the more than 2800 input utterances even though only 50% of these inputs were correctly recognized word for word. The average speech rate by subjects was 2.9 sentences per minute; the average task completion times for successful dialogs were 6.5 minutes. After the formal experiment, the system was run on a Sparc-2 workstation where the average response time by the computer was 2.2 seconds.[5]

Finally, general differences in user behavior depending on the level of computer initiative were also observed. When the computer operated in declarative mode, thus yielding the initiative to experienced human users, the dialogs

- were completed faster (4.5 minutes versus 8.5 minutes).
- had fewer user utterances per dialog (10.7 versus 27.6).

[2] Quintus Prolog is a trademark of Quintus Computer Systems, Incorporated.

[3] DECtalk is a trademark of Digital Equipment Corporation.

[4] Of the 23 dialogs which were not completed, 22 were terminated prematurely due to excessive time being spent on the dialog. 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 dialog.

[5] The system software as well as the experiment transcripts are available via anonymous FTP. Contact the author for details.

Subdialog Type	Directive Segment Utterances	Declarative Segment Utterances
Introduction	1-4	1-2
Assessment	5-14	3
Diagnosis	15	4-5

Table 1: Utterance classification into major subdialogs

- had users speaking longer utterances (63% of the user utterances were multi-word versus 40% in directive mode).

5. DIFFERENCES IN DIALOG CHARACTERISTICS

5.1. Major Subdialogs in Task Assistance

For task-oriented dialogs Grosz, 1978 has noted that *the structure of a dialog mirrors the structure of the underlying task*. Moody, 1988 conducted a Wizard of Oz study on the effects of restricted vocabulary on interactive spoken dialog. Her data were the basis for the formulation of the experimental Circuit Fix-It Shop system. From her analysis, five primary subdialog tasks were identified:

- Introduction Subdialog—establish the purpose of the task (e.g. to fix the circuit with ID number RS111).
- Assessment Subdialog—establish the current system behavior.
- Diagnosis Subdialog—establish the cause for the errant system behavior.
- Repair Subdialog—establish that the correction for the errant behavior has been made.
- Test Subdialog—establish that the system behavior is now correct.

Table 1 shows the classification of the utterances from the sample dialog segments shown previously into the various subdialogs. The continuation of these dialog segments that show the Repair and Test subdialogs have been omitted. Based on this general classification of repair dialogs, the following questions were investigated:

1. Were there differences in the relative number of utterances spoken in each subdialog as a function of the computer's initiative level?
2. Were there differences in the method of transitioning from one subdialog to the next as a function of the computer's initiative level?
3. What was the impact of miscommunication on the dialog structure?

The remainder of the paper presents the results of these investigations.

5.2. Utterance Classification

Hypotheses Since a variable initiative dialog allows the computer to yield dialog control to the human user as the user gains expertise, it is expected that the experienced user will exploit the situation to restrict the dialog to specific issues of interest. Presumably, an experienced user has substantial knowledge about the general behavior of the circuit, the basic nature of repairs, and how to determine when it is working, but will need some assistance with a specific problem diagnosis. Consequently, we would expect the following differences between modes as the computer yields initiative to the user:

- Introduction Subdialog—The number of utterances will change little.
- Assessment Subdialog—The number of utterances will be reduced in declarative mode.

Subdialog Type	Directive Mode Average	Directive Mode Percent	Declarative Mode Average	Declarative Mode Percent
Introduction	3.0	5.3%	3.1	11.1%
Assessment	15.9	28.2%	9.9	35.3%
Diagnosis	13.7	24.3%	8.5	30.4%
Repair	2.6	4.6%	0.4	1.4%
Test	21.2	37.6%	6.1	21.8%

Table 2: Utterance breakdown into major subdialogs

Subdialog Type	Directive Mode	Declarative Mode
Introduction	0.0%	0.0%
Assessment	10.5%	21.3%
Diagnosis	0.0%	7.0%
Repair	0.0%	61.1%
Test	23.5%	86.7%

Table 3: Percentage of subdialogs initiated by the user

- Diagnosis Subdialog—The number of utterances changes little.
- Repair Subdialog—It probably depends on the task. If the repair process is basically the same once the error is diagnosed, few utterances will be required as repairs can be done automatically. If the repair process is highly dependent on the type of error, even the experienced user may require significant advice from the system.
- Test Subdialog—The number of utterances is very few (i.e. experienced users can verify functioning without dialog).

Results Table 2 shows the average and relative number of utterances spoken in each of the main subdialogs of the task. The reported data combines both computer and user utterances. The results tend to support the hypotheses.[6] Note that very few utterances were ever spoken during the Repair phase of declarative mode dialogs. This was 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. The major source of the reduction in the number of utterances spoken per dialog occurred in the Assessment and Test phases, especially the Test phase.

5.3. User Initiation of Subdialog Transitions

Another important feature of the dialog structure concerns the transitioning between subdialogs. When the computer has total control of the dialog by operating in directive mode, it is expected that the computer will initiate the transitions between each subdialog. 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 subdialog to the next. The user controls the dialog but still requires computer assistance. Consequently, it would be expected that the computer would still initiate many of the transitions to the Assessment and Diagnosis phases, but that the user will be able to transition to other subdialogs 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 knowledge of proper functioning would be associated

[6] Because of the variability in individual subject behavior and problem situation, we cannot claim that the differences are statistically significant over the entire range of problems.

with a more experienced user. The results shown in table 3 provide support for these claims.[7] When the computer had the initiative (the directive mode dialogs), very few subdialog 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 dialogs), users initiated transition to the final stage of the dialog almost every time. In the intermediate stages, the computer still initiated most subdialogs, but users occasionally felt compelled to cause a change to a different phase. This virtually never happened when the computer had the initiative. Not counting the Introduction, which had to be initiated by the computer, only 10% of all subdialogs in directive mode were initiated by the user while 35% of the subdialogs in declarative mode were user-initiated.

5.4. The Impact of Miscommunication

One important phenomenon of interactive dialog that has not received much attention is the handling of miscommunication.[8] In the Circuit Fix-It Shop computer misunderstanding of user utterances occurred 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. 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 is filled with misunderstanding, so we should expect it in human-computer dialog as well. For the current system, how did miscommunication impact on the dialog structure?

Experimenter Interaction Computer misinterpretation of user utterances due to misrecognition of words can cause confusion between the user and computer and ultimate failure of the dialog. For the experiment, whenever a serious misrecognition occurred that caused the computer to interpret the utterance in a way which contradicted what was meant, the experimenter was allowed to tell the subject that a misrecognition had occurred, and to tell the user 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 user, "Due to misrecognition, your words came out as 'faster'." It is important to note that the experimenter did not tell the user what to do, but merely described what happened. In this way, the interaction was restricted to being between the computer and user as much as possible, given the quality of commercial, real-time, continuous speech recognition devices at the time of the experiment. With the computer running in declarative mode, the experimenter had to make such statements after 11.7% of the user's utterances, but when the computer was operating in directive mode, such statements were required after only 3.8% of the user's utterances. Not all misrecognitions required experimenter interaction (only about 24% of them in directive mode, and 41% of them in declarative mode).

User Initiated Corrections The previously mentioned paradigm 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 initiate a transition to a different subdialog in order to redirect the computer's focus when an error situation occurs. Conversely, when the machine has the initiative it 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. Consequently, users will give up trying to redirect the computer's attention when the computer has the initiative. This is borne out by the results. In directive mode, only 12% of all user-initiated subdialog transitions were attempts at error correction while in declarative mode, 26% of all user-initiated transitions were attempts at error correction.[9]

[7] These percentages are based on the following formula: (total number of subdialogs initiatied by the user) / (total number of different subdialogs of this type) * 100. For example, during the declarative mode dialogs there were 155 assessment subdialogs, 33 of which were initiated by the user.

[8] Exceptions are the work of McRoy and Hirst, 1993 on abductive interpretation of utterances in the detection of misunderstanding, and the work of Lambert and Carberry, 1992 on negotiation subdialogs that can be used for eliminating misunderstanding after it has been detected.

[9] Overall, while the computer was operating in directive mode, the user attempted to correct only 25% of the misunderstandings. In contrast, while the computer was operating in declarative mode, the user attempted to correct 54% of the

6. CONCLUSIONS

While there is ample analysis of dialog structure based on human-human and simulated human-computer dialog, there is very little information on the structure of actual human-computer dialog. In this paper we have overviewed how the Missing Axiom Theory for language use provides an integrated approach to dialog processing that allows a system to support variable initiative behavior in its interactions with human users. Based on the evaluation of this system at two different levels of initiative, we have observed the following phenomena:

- Directive mode dialogs tend to follow an orderly dialog pattern consisting largely of computer-initiated subdialog transitions and terse user responses. However, the inflexibility of this mode is a severe drawback in the presence of user-correctable miscommunications. A mechanism that allows the computer to change initiative within a dialog is needed.[10]
- Declarative mode dialogs are less orderly, consisting of more user-initiated subdialog transitions and provide 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 dialog system is to be useful in practical applications.

The observations reported in this paper are a first step in the long road to a comprehensive model of actual human-computer dialog structure. It is hoped that these results encourage other researchers constructing experimental systems to sufficiently test their systems and report on the results so that a more comprehensive view of human-computer dialog structure can be obtained.

7. ACKNOWLEDGMENTS

The author wishes to express his thanks to Steven A. Gordon who assisted with the transcript encoding and data analysis. Other researchers who have 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 Number NSF-IRI-9501571.

REFERENCES

Allen, J.F., Schubert, L.K., Ferguson, G., Heeman, P., Hwang, C.H., Kato, T., Light, M., Martin, N., Miller, B., Poesio, M., and Traum, D.R. 1995. The TRAINS project: a case study in building a conversational planning agent. *Journal of Experimental and Theoretical Artificial Intelligence*, 7:7–48.

Carberry, S. 1990. *Plan Recognition in Natural Language Dialogue.* MIT Press, Cambridge, Mass.

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

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

Guinn, C.I. 1994. *Meta-Dialogue Behaviors: Improving the Efficiency of Human–Machine Dialogue — A Computational Model of Variable Initiative and Negotiation in Collaborative Problem-Solving.* Ph.D. thesis, Duke University.

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

Lambert, L. and Carberry, S. 1992. Modeling negotiation subdialogues. In *Proceedings of the 30th Annual Meeting of the Association for Computational Linguistics*, pages 193–200.

misunderstandings. Not all attempts at error correction required a transition to a new subdialog.

[10] See Guinn, 1994 for ongoing work in this area.

McRoy, S. and Hirst, G. 1993. Abductive explanation of dialog misunderstandings. In *Proceedings of the Sixth Conference of the European Chapter of the Association for Computational Linguistics*, pages 277–286.

Moody, T. S. 1988. *The Effects of Restricted Vocabulary Size on Voice Interactive Discourse Structure.* Ph.D. thesis, North Carolina State University.

Oviatt, Sharon L. and Cohen, Philip R. 1991. Discourse structure and performance efficiency in interactive and non-interactive spoken modalities. *Computer Speech and Language*, 5:297–326.

Smith, R.W. and Hipp, D.R. 1994. *Spoken Natural Language Dialog Systems: A Practical Approach.* Oxford University Press, New York.

Smith, R.W., Hipp, D.R., and Biermann, A.W. 1995. An architecture for voice dialog systems based on Prolog-style theorem-proving. *Computational Linguistics.* In press.

Smith, R.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.

Walker, M. and Whittaker, S. 1990. Mixed initiative in dialogue: An investigation into discourse segmentation. In *Proceedings of the 28th Annual Meeting of the Association for Computational Linguistics*, pages 70–78.

Young, S.J. and Proctor, C.E. 1989. The design and implementation of dialogue control in voice operated database inquiry systems. *Computer Speech and Language*, 3:329–353.

Young, S.R., Hauptmann, A.G., Ward, W.H., Smith, E.T., and Werner, P. 1989. High level knowledge sources in usable speech recognition systems. *Communications of the ACM*, pages 183–194, February.