Effective Spoken Natural Language Dialog Requires Variable Initiative Behavior: An Empirical Study

Ronnie W. Smith
Department of Mathematics
East Carolina University
Greenville, NC 27858
email: rws@cs.duke.edu

Abstract

The usefulness of spoken natural language dialog for human-computer interaction is unquestioned. A spoken natural language interface provides a flexible means of communication while enabling human users to keep their hands and eyes busy on the task at hand. In this paper we claim that spoken natural language dialog is even more useful if the participating system permits variable initiative behavior. This is behavior where the task initiative can vary from strongly computer controlled to strongly user controlled or somewhere in between. Such behavior allows a system to effectively communicate with both task novices and experts as well as users with intermediate levels of expertise. In support of this claim, we present experimental results from a study of an implemented system that can operate at differing levels of initiative while engaging in spoken natural language dialog with users in order to provide assistance in repairing an electronic circuit. The results are based on eight users who participated in a total of 141 dialogs speaking 2840 utterances.

1 Integrated Dialog Processing for Natural Language Systems

Several researchers have built dialog systems to study various aspects of dialog processing including quantification, ellipsis, anaphoric reference, plan recognition, and user model. Examples of these systems are described in [Allen *et al.*, 1982], [Bobrow *et al.*, 1977], [Carberry, 1990], [Frederking, 1988], [Hendrix *et al.*, 1978], [Hoeppner *et al.*, 1983], and [Wilensky *et al.*, 1988]. However, there is little work on integrating the various aspects of dialog processing into a unified whole (see [Allen *et al.*, 1989] and [Young *et al.*, 1989]). Consequently, we developed a dialog processing model ([Smith *et al.*, 1992]) that when implemented in an electronic repair domain exhibits a number of behaviors required for efficient human-machine dialog including: (1) problem-solving; (2) coherent subdialog movement; (3) user model usage; (4) expectation usage; and (5) variable initiative behavior.

```
         A
       / | \
      B  C  D
     / \
   B1   B2
```

Figure 1: Sample Proof Tree

The key aspect of our model that enables the integration of various dialog processing mechanisms is the Missing Axiom Theory for driving the dialog process. The first step in integrating the required domain reasoning with a general natural language dialog mechanism is to include a reasoning system for achieving goals. We chose to use theorem proving, a well understood computational approach. The Missing Axiom Theory says to utilize language interaction with the user to acquire the missing axioms needed for proving that a target goal has been accomplished. 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 subogal B which leads to subgoals B1 and B2. Further suppose that B1 and B2 are domain facts not ordinarily available to the machine. Our system will resort to natural language interaction with the user to try to verify them. For example, B1 might require that the user know how to locate an item and B2 might require the user to report on that item's status. These are known as "missing axioms" and could yield the following dialog (C denotes the computer and U the user):

```
C: The meter is at the top of the blue region.
U: I see it.
C: What is it reading?
U: Three point five.
```

Completion of this dialog enables the proof of B and possibly, at a later time A.

This example also illustrates how user modeling is done. For example, if the user had referenced the meter in an earlier interaction, then B1 would already 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 meter.

The Missing Axiom Theory provides for a close correlation between subdialogs and domain goals. A subdialog consists of all the language interaction pertaining to a domain goal. Clarification subdialogs can be processed via dynamic modification of the relevant theorem. Movement between and within subdialogs is based on determining the domain goal to which a user input is relevant. This requires using *expectations* about the user's next utterance. These expectations are based on the current set of active goals. 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] and [Litman and Allen, 1987]). Movement to a goal other than the currently active one is termed an *interrupt* of the current subdialog. The way in which the computer responds to interrupts is a key factor in enabling variable initiative dialog to occur. This is described in the next section.

2 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 of the dialog and accomplish several goals without much computer assistance while at other times, the user 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 has in the dialog. These are described below:

1. directive - The computer has complete dialog control. It recommends a task goal for completion and will use whatever dialog is necessary to complete this goal. No interruptions to other subdialogs are allowed.
2. suggestive - The computer still has dialog control, but not as strongly. The computer will recommend a task goal for completion, but will allow minor interruptions to closely related subdialogs.
3. declarative - The user has dialog control and can interrupt to any desired subdialog 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 dialog control. Consequently, the computer will passively acknowledge user statements. It will provide information only as a direct response to a user question.

One observes that the degree of interruptibility allowed by the computer increases from directive to passive mode. Thus, the dialog mode has a critical effect on the computer's choice of response, particularly in an interrupt situation. In fact, this selection process is a function of three primary factors: (1) the current computer goal; (2) the current user focus; and (3) the current dialog mode. Details about this selection process are provided in [Smith, 1991]. This process has been implemented as part of a spoken natural language dialog system that assists users in the repair of an electronic circuit. The two dialog segments of figure 2, obtained from actual usage of the implemented system, illustrate the differences between mode for the two modes in which we experimentally evaluated the system: directive and declarative. 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 include [Whittaker and Stenton, 1988], [Walker and Whittaker, 1990], and [Kitano and Van Ess-Dykema, 1991]. In addition to a theory, we provide a mechanism that allows a system to participate in such dialogs.

3 Implementation

The implemented dialog system is running on a Sparc II workstation with the majority of the code written in Quintus Prolog[1], and the parser in C. The

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

```
Dialog Segment 1 (Directive Mode)

C: This is the circuit fix it shop.  How may I
      help you?
U: I want to fix a circuit.
C: What is the id of the circuit?
U: Rs111.
C: I am familiar with that circuit. The LED is
      supposed to be displaying alternately
      flashing one and seven.
U: The LED is off.
C: What is the switch at when the LED is off?
U: The switch is down.
C: Put the switch up.
U: Okay.
C: Put the knob to one zero.
U: Okay.
C: What is the LED displaying?
U: LED is displaying alternately flashing one
      and top of seven.
C: Is there a wire between connector one zero
      two and connector one zero four?

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

Dialog Segment 2 (Declarative Mode)

C: This is the circuit fix it shop.  How may I
      help you?
U: Rs111 is not working.
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.

U: There is no wire on one zero four.
C: There is supposed to be a wire between
      connector one zero four and connector
      one zero two.

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

Figure 2: Sample dialog segments: directive and declarative

implemented domain processor provides information about 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.

For the dialog system, speech recognition is performed by a Verbex 6000 running on an IBM PC and speech production is performed by a DECtalk[2] DTCO1 text-to-speech converter. The users are restricted to a 125 word vocabulary in the connected speech system.

4 Evaluation: Problem-Solving Effectiveness

The system was experimentally evaluated using eight Duke University undergraduates. These students had completed one computer science class and were taking a second, but none had taken any AI classes nor had they previously used a natural language system. Furthermore, none were electrical engineering students, who could have probably fixed the circuit without any computer assistance. After a training session, each subject participated in two problem-solving sessions, one in directive mode, where the computer had maximal initiative, and one in declarative mode, where the user had the initiative. Because we have not yet developed a good strategy for automatically changing modes during a dialog, we had to lock the system in one mode for the duration of a dialog to study the effects of different levels of initiative.

In each of the problem-solving sessions a subject could work on up to ten different problems. A total of 141 problems were attempted of which 118 or 84% were successfully completed. Of the 23 that were not completed successfully, 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. The domain processor did not have sufficient knowledge to properly assist the user in this situation. 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. A detailed description of the experimental procedure and results is given in [Smith, 1991].

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

5 Evaluation: The Need for Variable Initiative

5.1 Exploiting Expertise

Analysis of the transcripts obtained in the 141 attempted dialogs indicate that as users gain expertise and are allowed to direct the dialog that they tend to

1. exploit their expertise to carry out several steps independent of any dialog interaction;
2. communicate with the system only when general advice is needed; and
3. specify more completely the domain situation in order to provide the proper context to the system.

This analysis is supported by the empirical measurements listed below in table 1.

	System Mode	
	Declarative	Directive
avg. completion time	303.8	541.0
avg. # utterances/dialog	12.0	28.8
% non-trivial utterances	62.9	39.7
avg. subject response time (sec.)	17.0	11.8

Table 1: Basic Performance Results

A description of the parameters is given below:

- average completion time - the average time (in seconds) required to complete a dialog. This average is computed for all successful dialogs concerning any of the first eight problems of a session (the first eight problems of the problem-solving sessions were balanced for difficulty).
- average # of utterances/dialog - the average number of subject utterances per dialog for all completed dialogs concerning any of the first eight problems of a session.
- % non-trivial utterances - the percentage of subject utterances in a session which were "non-trivial" (i.e. consisted of more than one word).
- average subject response time - gives the average elapsed time (in seconds), from the computer's utterance to the subject's response.

The results provide an indication that after the user has gained some experience and has more initiative: (1) the average completion time decreases; (2) the average number of utterances decreases; (3) the percentage of non-trivial utterances increases; and (4) the elapsed time between computer utterance and subject response increases.

These results are consistent with the intuition that more experienced users are better able to take charge of the dialog and complete the task more quickly, using fewer utterances, but providing more information per utterance and spending more time in thought before producing an utterance. A variable initiative system can be beneficial to experienced users.

5.2 Favorable User Perceptions

The previous section supports the intuitive idea that experienced users can more efficiently complete the task when given more initiative, but can a variable initiative system provide the right level of control for both the experienced user and the beginner? An exit survey completed by our experimental subjects provides a favorable answer. As part of the survey, subjects were asked to indicate their level of agreement with the following statements:

```
"At the start of the experiment the system
    had too much control of the dialog."

"At the end of the experiment the system
    had too much control of the dialog."
```

Subjects rated their level of agreement with these statements by circling the appropriate number on the following scale: 1-highly disagree; 2-disagree; 3-do not know; 4-agree; 5-highly agree. The average rating for the statement concerning too much control at the beginning was 2.8. However, for the statement concerning too much control at the end, subjects operating in declarative mode in the final session (where the user had the initiative) gave an average rating of 2.5 while subjects operating in directive mode (strong computer initiative), gave an average rating of 4.1.

Thus, while the subjects who had control at the end were content with the system interaction, the subjects who were forced to give back control to the computer in the final session agreed that the computer had too much control at the end. We see from these results that a variable initiative system can provide the proper level of control for both the beginning and the experienced user.

6 Evaluation: System Requirements for Variable Initiative

In the previous two sections we have seen experimental results that support the idea that a variable initiative system is capable of providing the flexibility neeeded to interact successfully with both experienced and novice users. The results also provide an indication of the types of knowledge and processing a computer system needs to enable this flexible interaction. They include

- a model of the user's goals/focus;
- a rich domain model; and
- robust behavior in the presence of speech recognition errors.

These factors are discussed below.

6.1 A Model of the User

If the system is to allow users to have the initiative at various times in a dialog, and initiative is related to the ability of the user to interrupt to his or her own goals of interest, a cooperative system must be able to identify and track what a user's goals are (or could possibly be) in order to respond appropriately. Our system did not perform explicit plan recognition but relied on identification of the subcircuit to which a user's utterance pertained in order to provide a cooperative response. Explicit recognition of a user's goals should be more crucial as the tight coupling between task and dialog is relaxed. Note that the flexibility of dialog can allow a system some leeway. If it does not identify the exact user goal the first time, the user has the ability to more directly inform the computer of the explicit goal of interest (e.g. "I want to examine the power circuit.")

6.2 A Rich Domain Model

In support of obtaining a model of the user's goals, the general dialog system must have access to a rich model of the problem domain. Our architecture supports such access for a general dialog processor via a standardized interface to the domain processing module (see [Smith, 1991] for details). This domain-dependent module must be capable of providing the needed types of reasoning as requested by the dialog processor.

In particular, when there is an attempt to complete a domain goal, and there is a missing axiom that requires the use of language, the domain processor must provide the semantic expectations pertaining to explicit domain observations (e.g. value(switch,position,up)). In contrast, the dialog processing component describes expectations based on general characteristics about goals in task-oriented dialogs and includes any utterances where context is required to complete the interpretation. Examples of such expectations include yes/no responses, clarification requests, elliptical responses, and pronominal reference.

It was previously stated that when the user has the initiative, the utterances tend to more completely specify the domain situation. This is borne out empirically by an examination of the 2840 utterances collected during the experiment. After inserting discourse monitoring statements into the system software and rerunning simulations of the original dialogs, we observed that when the user had the initiative that domain knowledge was the expectation source for the user utterance 67% of the time, but only 32% of the time when the computer had the initiative. These results support the idea that a variable initiative system requires a rich domain model so that domain specific expectations can be used by the general dialog system.

6.3 Robust Behavior

Finally, variable initiative behavior provides an interesting tradeoff when spoken interaction is allowed. User initiative encourages longer user utterances that increase the likelihood of a misrecognition and possibly a misunderstanding, but user initiative also facilitates error recovery from miscommunication. [Hipp, 1992] describes a robust parsing algorithm that always finds a grammatical parse of an utterance, although it may require additions, deletions, and/or substitutions of words. This algorithm, which uses dialog expectations to assist in the interpretation process, enabled our system to correctly understand 81.5% of all utterances although only 50.0% were correctly recognized word for word by the Verbex speech recognizer. Nevertheless, 18.5% of the utterances were not fully interpreted correctly.

These misinterpretations can cause confusion between the user and system and ultimate failure of the dialog. For the experiment, whenever a serious misrecognition occurred that caused the system 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 system, but *could say nothing else.* With the system running in declarative mode, the experimenter had to make such statements after 11.7% of the user's utterances, but when the system was operating in directive mode, such statements were required after only 3.8% of the user's utterances. Not all misrecognitions required experimenter interaction.

This paradigm for notifying the user of a serious misrecognition leaves the responsibility with the user to try to correct the computer's misunderstanding. User correction is much easier to do with user initiative as the user can simply repeat the previous utterance, and the computer will allow an "interrupt" back to the original goal. Unfortunately, when the computer has the initiative, it might proceed down the wrong problem-solving path for many utterances before questioning the erroneous statement. Thus, user initiative enhances the chance for miscommunication in a given utterance, but it also enhances the opportunity to quickly recover from such miscommunications.

It is unlikely that speech recognizers will be 100% accurate anytime in the near future, and robust mechanisms for recovery from miscommunication are essential. Variable initiative behavior provides a natural one — if necessary, allow the user to take the initiative to correct the misunderstanding before resetting the initiative to the original value. However, an algorithm for automatically varying initiative remains an open problem.

7 Acknowledgments

This research was supported by National Science Foundation Grant Number NSF-IRI-88-03802, and by Duke University and East Carolina University. Other researchers who have contributed to this project include Alan W. Biermann, Robert D. Rodman, Ruth S. Day, D. Richard Hipp, Dania Egedi, and Robin Gambill.

References

[Allen *et al.*, 1989] J. Allen, S. Guez, L. Hoebel, E. Hinkelman, K. Jackson, A. Kyburg, and D. Traum. The discourse system project. Technical Report 317, University of Rochester, November 1989.

[Allen *et al.*, 1982] J.F. Allen, A.M. Frisch, and D.J. Litman. ARGOT: The Rochester dialogue system. In *Proceedings of the 2nd National Conference on Artificial Intelligence*, pages 66–70, 1982.

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

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

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

[Hendrix *et al.*, 1978] G. G. Hendrix, E.D. Sacerdoti, D. Sagalowicz, and J. Slocum. Developing a natural language interface to complex data. *ACM Transactions on Database Systems*, pages 105–147, June 1978.

[Hipp, 1992] D.R. Hipp. *Design and Development of Spoken Natural-Language Dialog Parsing Systems.* PhD thesis, Duke University, 1992.

[Hoeppner *et al.*, 1983] W. Hoeppner, T. Christaller, H. Marburger, K. Morik, B. Nebel, M. O'Leary, and W. Wahlster. 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 Aritificial Intelligence*, pages 588–594, 1983.

[Kitano and Van Ess-Dykema, 1991] H. Kitano and C. Van Ess-Dykema. 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, 1991.

[Litman and Allen, 1987] D.J. Litman and J.F. Allen. A plan recognition model for subdialogues in conversations. *Cognitive Science*, 11(2):163–200, 1987.

[Smith, 1991] R.W. Smith. *A Computational Model of Expectation-Driven Mixed-Initiative Dialog Processing.* PhD thesis, Duke University, 1991.

[Smith *et al.*, 1992] R.W. Smith, D.R. Hipp, and A.W. Biermann. A dialog control algorithm and its performance. In *Proceedings of the 3rd Conference on Applied Natural Language Processing*, pages 9–16, 1992.

[Walker and Whittaker, 1990] M. Walker and S. Whittaker. 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, 1990.

[Whittaker and Stenton, 1988] S. Whittaker and P. Stenton. Cues and control in expert-client dialogues. In *Proceedings of the 26th Annual Meeting of the Association for Computational Linguistics*, pages 123–130, 1988.

[Wilensky *et al.*, 1988] R. Wilensky, D.N. Chin, M. Luria, J. Martin, J. Mayfield, and D. Wu. The Berkeley UNIX consultant project. *Computational Linguistics*, 14(4):35–84, 1988.

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

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