Efficient Collaborative Discourse: A Theory and Its Implementation

Alan W. Biermann, Curry I. Guinn, D. Richard Hipp, Ronnie W. Smith

Computer Science Department
Duke University
Durham, NC 27706

ABSTRACT

An architecture for voice dialogue machines is described with emphasis on the problem solving and high level decision making mechanisms. The architecture provides facilities for generating voice interactions aimed at cooperative human-machine problem solving. It assumes that the dialogue will consist of a series of local self-consistent subdialogues each aimed at subgoals related to the overall task. The discourse may consist of a set of such subdialogues with jumps from one subdialogue to the other in a search for a successful conclusion. The architecture maintains a user model to assure that interactions properly account for the level of competence of the user, and it includes an ability for the machine to take the initiative or yield the initiative to the user. It uses expectation from the dialogue processor to aid in the correction of errors from the speech recognizer.

1. Supporting the Voice Technologies

Dialogue theory is the implementing science for the voice technologies. The many successes in voice recognition and generation will have value only to the extent that they become incorporated into practical systems that deliver service to users. This paper reports on a dialogue system design that attempts to implement a variety of behaviors that we believe to be necessary for efficient human-machine interaction. These behaviors include:

1. Collaborative problem-solving: The system must have the ability for the machine to problem-solve and collaborate with the human user in the process. Specifically, the machine must be able to formulate queries to the user and process responses that will enable progress toward the goal.
2. Subdialogue processing: It must be able to participate in locally coherent subdialogues to solve subgoals and to jump in possibly unpredictable ways from subdialogue to subdialogue in an aggressive search for the most effective path to success. Such jumps may emanate from the system's own processing strategy or they may be initiated by the user and tracked through plan recognition by the system.
3. User modeling: It needs to maintain a user model that enables it to formulate queries appropriate to the user and that will inhibit outputs that will not be helpful.
4. Variable initiative: The machine must be able to take the initiative and lead the interaction at places where it has information implying that it can do this effectively. It also needs to be able to yield the initiative completely or in part at times when data is available indicating that it should do so. It needs to be able to negotiate with the user to either take or release the initiative when appropriate.
5. The use of expectation: It needs to be able to use the expectation implicit in the dialogue to support the voice recognition stage in error correction.

2. A Dialogue System Architecture

Despite the variety of the target behaviors and their seeming structural disjointness, an architecture has been found that supports them all in a relatively uniform and natural way [1, 2, 3, 4]. The design is based on the model of a Prolog processor but includes a variety of special capabilities to address the needs of this application. This section will describe the fundamental theory of the system and the next section will describe its performance in a series of tests with human subjects.

The basic operation of the architecture is illustrated in Figure 1 where problem-solving is to achieve top level goal G. Prolog-style theorem proving proceeds in the usual way and if G can be proven from available information there will be no interaction with the user. However, if information is not sufficient to allow completion of the proof, the system can attempt to provide "missing axioms" through interaction with the user. In the figure, this process is illustrated in the subtree C where P has been proven from an existing assertion but Q is not known. Then the system may be able to resort to a voice interaction with the user to discover Q. Thus the architecture organizes interactions with the user to directly support the theorem proving process. This organization gives the dialogue the task-oriented coherent ([5]) organization that is needed for effective cooperative problem-solving. It provides the **intentional structure** described by Grosz and Sidner[6].

The example continues with the illustrated rule

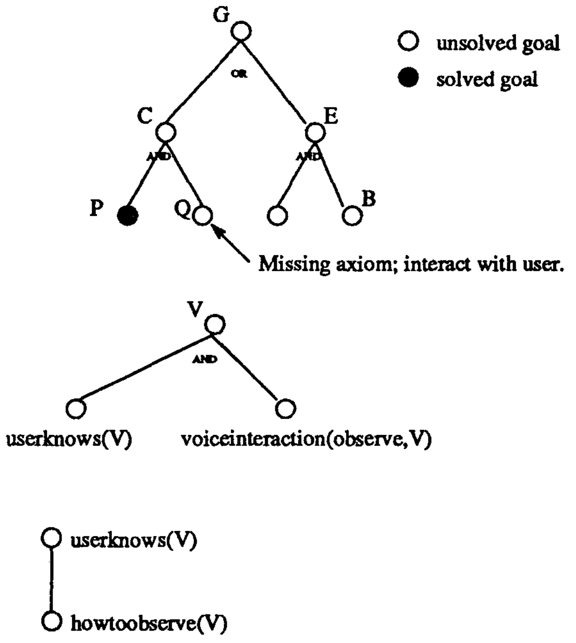

Figure 1: The theorem proving tree associated with a voice dialogue.

V :- userknows(V), voiceinteraction(observe,V)

which is also shown in Figure 1. Specifically, it asserts that if, according to the user model[7, 8, 9, 10], the user knows V, then a voice interaction could be initiated to try to obtain that information. Our approach effectively enables V to unify with any goal to enable the interaction. This could yield an exchange between computer and user of the type

C: Is the switch on?
U: Yes.

But the situation might not be as simple as a single question and answer. It may be that the user does not know how to observe Q but could be told. This is illustrated by the rules

userknows(V) :- howtoobserve(V)
howtoobserve(V) :- . . .

which could lead to a lengthy interaction involving locating other objects, carrying out actions, and making other observations. Thus a series of voice interactions could ensue with the goal of eventually observing Q. The set of all interactions aimed at the completion of a given goal is defined by this project to be a **subdialogue**. Notice that the subdialogue accounts at every step for the user's knowledge through invocation of the user modeling assertions. The dialogue asks only questions that the user model indicates are appropriate and explains concepts either extensively, briefly, or not at all depending on the assertions contained in the model. Subdialogues by one name or another have been studied by a variety of authors [11, 12, 13, 14].

The system allows for the possibility of unpredictable jumps from one subdialogue to another. In the above example, the user might be locally uncooperative and respond as follows:

C: Is the switch up?
U: B is true.

Here we assume that B is an assertion related to another subgoal on the theorem proving tree as shown in Figure 1. The user may initiate such a change in subdialogue in an attempt to pursue another path to the global goal. Here the machine first must track the user's intention (in a process called "plan recognition" [15, 16, 17, 18, 19]) and then evaluate whether to follow the move or not. This decision is based upon the current level of the initiative of the system as described below. If the system follows the user's initiative, it will apply its internal theorem proving system to the subgoal E and pursue voice interactions related to it. If it rejects the user's indicated path, it will simply store the received fact and reaffirm its own path:

C: Is the switch up?

The system may also abandon a subdialogue for reasons of its own. For example, processing during the dialogue could yield the unexpected result that the current path is no longer likely to yield an efficient path to the global goal. Then the system could abruptly drop a line of interactions and jump to a new subgoal which is momentarily evaluated as more attractive.

Efficient dialogue often requires regular changes of initiative depending on which participant currently has the key information[20, 21, 22, 23]. When a subject is opened where one participant is knowledgeable and the other is not, that participant should lead the interaction to its completion and the other should be supportive and respond cooperatively. Our project implements four levels of initiative, directive, suggestive, declarative, and passive. These levels result in, respectively, uncompromising control on the part of the machine, control but only at a weaker level, the yielding of control to the user but with a willingness to make assertions about the problem-solving process, and quiet acceptance of the user's initiative. The level of initiative sets the strength at which the machine will prefer its own best evaluated solution path when it selects the subdialogue to be followed. The initiative level also adjusts the assertiveness of the spoken outputs and may affect the way inputs are processed. (See [1]).

Expectation at each point in a dialogue is derived from the proof tree and other dialogue information in a manner similar to that explained by Young[24]. Concepts that would be appropriate in the context of the current local interaction are "unparsed" into expected syntactic inputs and voice recognition is biased to receive one of these expected inputs. If the recognition phase fails to achieve a good match with a local expectation, comparisons are made to nonlocal expectations at increasing distances from the local context until an acceptable match is found or an error message is reported. Recognition of a nonlocal expectation amounts to the discovery that the user is following a different path; this is a process called "plan recognition" in the literature. If the system is following the user initiative at this point, it may shift its theorem proving efforts to that subtree and cooperate with the user.

3. The Implementation

The major system developed by this project is known as "The Circuit Fix-It Shoppe" [1, 25]. It is implemented with a domain modeller to guide the process of debugging an electric circuit and to present appropriate subgoals for possible examination. A complex dialogue controller overviews the processing of decisions related to which subgoal to select and level of initiative issues.

The coding has been done primarily in Quintus Prolog on a Sun 4 workstation. The parser is coded in C and uses a nearest neighbor algorithm to find the best match between the error prone recognition system output and the set of expected inputs. The speech recognizer is a Verbex 6000 user-dependent connected speech recognizer running on an IBM PC, and the vocabulary is currently 125 words. Users are required to begin each utterance with the word "verbie" and end with the word "over."

The implemented domain processor is loaded with a model of a particular experimental circuit that was constructed on a Radio Shack 160-in-One Electronic Project Kit. The model is complete enough to do any problem of the circuit that involves missing wires. For example, if the system is asked to debug the circuit with no wires at all, it will systematically discover every missing wire and ask the user to install it.

Speech output is done by a DECtalk DTC01 text-to-speech converter.

4. Experimental Results

The system was tested with human subjects solving missing wire problems. Subjects were trained to use the system and given some guidance on acceptable sentence syntax. They were required to register their pronunciation of all of the vocabulary words and they spoke 239 sentences to train the recognizer on coarticulation. They were provided with a list of vocabulary words and charts on a poster board suggesting implemented syntax. They were told not to address the experimenter during the test but the experimenter had a set of eleven standard messages that he or she would give to the subject under certain specified conditions. These included such things as "The system is ready for your next utterance" or "Please restrict your utterances to one sentence." On two later days, subjects were given up to ten problems per session, one session with the machine at the strongest initiative level and one at a moderately passive level.

A transcript of one such dialogue follows here. This interaction was in moderately passive mode so the system tends to suggest rather than demand action. The user inserted two missing wires during the interaction and made two changes in subdialogue (marked as **) that were each correctly tracked by the system. The total time of the interaction was 160 seconds.

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.
C: The LED is supposed to be displaying 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 connector one one one and connector one two zero.
U: LED is displaying one and seven. **
C: Is the one on the LED flashing?
U: No.
C: Is the seven on the LED flashing?
U: No.
C: There is supposed to be a wire between connector three zero and connector seven zero.
U: RS111 is working. **
C: Glad to have been of assistance. Good-bye.

Eight subjects attempted 141 dialogues, 75 with the machine in mildly passive mode and 66 with it in directive mode. The average problem completion time in the two modes was 270.6 seconds and 511.3 seconds respectively. The longer sessions in the directive mode were because the system required the user to pedantically go through every step of a debugging procedure while in the more passive mode, the user could often jump to the correct subgoal and solve it quite quickly. The average number of utterances spoken per dialogue was 10.7 and 27.6, respectively. The experimenter needed to give error messages to the subject about one every six sentences with the machine in passive mode and one every eighteen sentences in directive mode. This indicates that with the greater freedom allowed by the more passive mode, subjects tended to get into more difficulty using the system. The exact sentence recognition rate by the Verbex machine in the two modes was 44.3 and 53.1 percents, respectively. These were corrected to 75.3 and 85.0 respectively by the expectation-based nearest neighbor error correction system.

5. Current Research

Our newest dialogue algorithm by Guinn[3] features a set of real numbers on the proof tree paths that are continuously updated to reflect estimates of the nearness to a solution. The algorithm follows paths using a best first strategy, and it includes automatic mechanisms to change mode, negotiate initiative, and other efficiency improving behaviors. This algorithm has not been incorporated into the voice interactive system and is instead being tested separately.

This algorithm allows a more complicated interaction to occur involving negotiation if the machine and user differ on who should control the initiative. Suppose the machine adamantly demands its own path (Is the switch up?) and the user is equally as uncompromising and demands information related to the E subgoal as shown in Figure 1. With Guinn's strategy the system negotiates with the user to try to convince the user to follow its path. Specifically, it presents the user with part of the proof tree leading to the goal to show the user how quickly the goal can be achieved. For example, in the case of Figure 1, it might assert

> C: If the switch is up, then since P is true, then C will be true; consequently G will be true.

Alternatively, the user could present his or her own path to the goal in a negotiation and conceivably convince the system to lower its evaluation of its own path.

This newer theory of initiative bases subdialogue decisions on a real number and biases the number with an initiative parameter which can take on any value between 0 and 1. In this system, the level of initiative is defined over a continuous range rather than a discrete set of initiative values.

Tests on the newer dialogue algorithm have been in machine-to-machine problem-solving sessions. The methodology has been to randomly distribute facts about a murder mystery between the two participants and then observe the conversations that lead to a solution of the mystery. The transmitted information between the participants is in the form of Prolog-style predicates since the machines gain nothing through a translation to natural language. Detailed results have been extremely encouraging and will be given later. For example, in one test involving 85 dialogues, the average number of interactions required to solve the problems was 123 without the negotiation feature described above and 103 with it.

6. Comparisons with Other Dialogue Systems

The system that most resembles the one we describe here is the MINDS system of Young *et al.* [26]. Their system maintains and AND-OR tree much like our Prolog tree and engages in dialogue similarly to try to achieve subgoals. It similarly uses expectations generated by subgoals and enhanced by a user model to predict incoming utterances for the purpose of error correction. The resulting system demonstrated dramatic improvements. For example, the effective perplexity in one test was reduced from 242.4 to 18.3 using dialogue level constraints while word recognition accuracy was increased from 82.1 percent to 97.0. We employ Prolog-style rules for the knowledge base and the associated proofs for directing the goal-oriented behavior. This leads to the "missing axiom theory" we describe above and some rather simple methods for handling the user model, multiple subdialogues, variable initiative, negotiation and a variety of other features.

Another dialogue system, by Allen *et al.* ([27]), uses a blackboard architecture to store representations of sentence processing and dialogue structures. Processing is done by a series of subroutines that function at the syntactic, semantic, and dialogue levels. This system models detailed interactions between the sentence and dialogue levels that are beyond anything we attempt but does not support problem-solving, variable initiative and voice interactions as we do.

A third interesting project has produced the TINA system[28]. This system uses probabilistic networks to parse token sequences provided by a speech recognition system, SUMMIT by Zue *et al.* [29]. The networks and their probabilities are created automatically from grammatical rules and text samples input by the designer. Their main utility is to provide expectation for error correction as we do in our system. However, their expectation is primarily syntax-based while ours uses structure from all levels, subdialogue (or focus-based), semantic and syntactic. Their semantics is built directly into the parse trees which is translated into SQL for access to a database. Our system is task-oriented, emphasizes problem-solving, and employs a user model to assure effectiveness of the interaction.

References

1. R.W. Smith, D.R. Hipp and A.W. Biermann. A Dialog Control Algorithm and its Performance. *Proc. of the Third Conf. on Applied Natural Language Processing*, Trento, Italy, 1992.
2. D.R. Hipp. *A New Technique for Parsing Ill-formed Spoken Natural-language Dialog*. Ph.D. thesis. Duke University, Durham, North Carolina. 1992.
3. C.I. Guinn. Ph.D. thesis. Duke University. Durham, North Carolina. To appear. 1993.
4. R.W. Smith. *A Computational Model of Expectation-Driven Mixed-Initiative Dialog Processing*. Ph.D. thesis, Duke University, Durham, North Carolina, 1991.
5. J.R. Hobbs. "Coherence and coreference." *Cognitive Science* 3:67–90, 1979.
6. B.J. Grosz and C.L. Sidner. Attentions, intentions, and the structure of discourse. *Computational Linguistics*, 12(3):175–204, 1986.
7. A. Kobsa and W. Wahlster, editors. *Special Issue on User Modeling*. MIT Press, Cambridge, Mass., September 1988. A special issue of *Computational Linguistics*.
8. R. Cohen and M. Jones. Incorporating user models into expert systems for educational diagnosis. In A. Kobsa and W. Wahlster, editors, *User Models in Dialog Systems*, pages 313–333. Springer-Verlag, New York, 1989.

9. T.W. Finin. GUMS: A general user modeling shell. In A. Kobsa and W. Wahlster, editors, *User Models in Dialog Systems*, pages 411–430. Springer-Verlag, New York, 1989.

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

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

12. C. Linde and J. Goguen. Structure of planning discourse. *J. Social Biol. Struct.* pages 1:219–251, 1978.

13. L. Polanyi and R. Scha. On the recursive structure of discourse. In *Connectedness in Sentence, Discourse and Text*, ed. by K. Ehlich and H. van Riemsdijk. Tilburg University. pages 141-178, 1983.

14. R. Reichman. *Getting Computers to Talk Like You and Me*. MIT Press, Cambridge, Mass., 1985.

15. J.F. Allen. Recognizing intentions from natural language utterances. In M. Brady and R.C. Berwick, editors, *Computational Models of Discourse*, pages 107–166. MIT Press, Cambridge, Mass., 1983.

16. H.A. Kautz. A formal theory of plan recognition and its implementation. in *Reasoning about Plans*, ed. by J.F. Allen, H.A. Kautz, R.N. Pelavin, and J.D. Tenenberg. San Mateo, California: Morgan Kaufmann, pages 69–125, 1991.

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

18. M.E. Pollack. A model of plan inference that distinguishes between the beliefs of actors and observers. In *Proceedings of the 24th Annual Meeting of the Association for Computational Linguistics*, pages 207–214, 1986.

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

20. 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.

21. D.G. Novick. *Control of Mixed-Initiative Discourse Through Meta-Locutionary Acts: A Computational Model*. PhD thesis, University of Oregon, 1988.

22. 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.

23. 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.

24. S.R. Young. Use of dialogue, pragmatics and semantics to enhance speech recognition. *Speech Communication*, 9:551–564, 1990.

25. D.R. Hipp and R.W. Smith. *A Demonstration of the 'Circuit Fix-It Shoppe'*. Twelve minute video tape, Department of Computer Science, Duke University, Durham, North Carolina. 1991.

26. 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.

27. 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.

28. S. Seneff. TINA: A Natural Language System for Spoken Language Applications. *Computational Linguistics*, 18(1):61-86, 1992.

29. V. Zue,J. Glass,M. Phillips and S. Seneff. The MIT SAUMMIT speech recognition system: a program report. *Proceedings, DARPA Speech and Natural Language Workshop*, Philadelphia, pages 21–23, 1989.

Acknowledgment

This research was supported by National Science Foundation grant number NSF-IRI-88-03802 and by Duke University.