Mechanisms for Mixed-Initiative Human–Computer Collaborative Discourse

Curry I. Guinn

Department of Computer Science
Duke University
Box 90129
Durham, NC 27708
`cig@cs.duke.edu`

Center for Digital Systems Engineering
Research Triangle Institute
Box 12194
Research Triangle Park, NC 27709
`cig@rti.org`

Abstract

In this paper, we examine mechanisms for automatic dialogue initiative setting. We show how to incorporate initiative changing in a task-oriented human–computer dialogue system, and we evaluate the effects of initiative both analytically and via computer–computer dialogue simulation.

1 Motivation: Initiative in Task-Oriented Dialogues

Efficient human–computer dialogue requires immediate utterance-by-utterance accommodation to the needs of the interaction. The participants must continuously focus all concerns on the goals of the task and avoid extraneous paths. This means that the **dialogue initiative** should always pass immediately to the participant who is best able to handle the current task. An agent is said to have *dialogue initiative* over a mutual goal when that agent controls how that goal will be solved by the collaborators. In the implemented voice dialogue system "The Circuit Fix-it Shop" (Smith et al., 1992; Smith and Hipp, 1995), the dialogue fragment given in Figure 1 occurs between a human user and a computer collaborator. Utterances 2 and 4 indicate that the computer is directing the search for the missing wire in the faulty circuit. However, in Utterance 5 the user takes the initiative and suggests a different wire than has been proposed by the computer. In this paper we will present a theory explaining how initiative changes between participants and how computational agents can evaluate who should be in control of solving a goal.

1	U	The LED is displaying an alternately flashing one and the top corner of a seven.
2	C	There is supposed to be a wire between connector nine nine and connector one zero zero.
3	U	It is there.
4	C	There is supposed to be a wire between connector nine eight and connector one zero two.
5	U	There is no wire on connector **one zero four**. [*emphasis mine*]
6	C	There is supposed to be a wire between connector one zero four and connector one zero two.

Table 1: A Dialogue Fragment in the Circuit Fix-it Shoppe Domain

2 Automating Dialogue Initiative

Many implemented dialogue systems are question-answer systems with fixed initiative where one agent is in control and the other agent is passive (the master-slave assumption). For instance, in the LADDER system (Hendrix et al., 1978) the user has the initiative while the computer is passive. In contrast, the VODIS system (Young and Proctor, 1989) has the computer taking the initiative while the user is passive. Some dialogue systems (like GUS (Bobrow et al., 1977)) allow for a more mixed-initiative interaction; however, the places where the user can take the initiative are limited and defined *a priori*. The dialogue model of Smith (1995) allows for either the computer or the user to assume degrees of initiative; however, Smith presents no algorithm for the computer to change initiative during a dialogue. Our model of mixed-initiative dialogue allows either participant to be in control of the dialogue at any point in time.

Like Smith (1995) we believe that *the level of initiative in the dialogue should mirror the level of initiative in the task* (which is a corollary to Grosz's

(1978) *the structure of a dialog mirrors the structure of the underlying task*). Unlike previous research in dialogue initiative, however, we attach an initiative level to **each goal** in the task tree. Thus an agent may have initiative over one goal but not another. As goals get pushed and popped from the problem-solving stack, initiative changes accordingly. Thus many initiative changes are done implicitly based on which goal is being solved.

2.1 The Setting of Initiative Levels

In our model of dialogue, initiative levels for each goal are defined during the interaction based on 1) explicit and implicit initiative-changing utterances and 2) competency evaluation.

Explicit and Implicit Initiative-Changing Utterances Several researchers (Whittaker and Stenton, 1988; Walker and Whittaker, 1990) have noted that dialogue control can be exchanged through overt cues in the discourse. Our model concentrates on two specific dialogue cues: questions and answers. When an agent A_1 asks another agent A_2 to satisfy a goal G, agent A_2 gains initiative over goal G *and all subgoals of* G *until agent* A_2 *passes control of one of those subgoals back to agent* A_1. A similar initiative-setting mechanism is fired if agent A_1 announces that it cannot satisfy goal G. When a goal has been answered (satisfied) the problem-solving stack is popped. The initiative will now belong to whomever the initiative is for the goal on top of the stack.[1]. In the following human-computer dialogue gathered by Smith and Hipp(Smith and Hipp, 1995),

U: How do I fix this circuit?
C: What is the LED displaying?
U: Nothing.
C: What is the switch at?
U: Where is the switch?
C: In the lower left corner.
U: The switch is down.
C: Put the switch up.

all initiative changes can be accounted for by explicit initiative-changing utterances or by popping of the problem-solving stack due to goal resolution as illustrated in Figure 1.

Competency Evaluation for Initiative Setting How does an agent decide whether to ask its collaborator for help? An obvious approach is to ask for help when the agent is unable to satisfy a goal on its own. This approach is the basic mechanism for several dialogue systems (Young et al., 1989; Smith and Hipp, 1995; Guinn, 1994). An additional approach is to ask the collaborator for help if it is believed that the collaborator has a better chance of solving the goal (or solving it more efficiently). Such an evaluation requires knowledge of the collaborating agent's capabilities as well as an understanding of the agent's own capabilities.

Our methodology for evaluating competency involves a probabilistic examination of the search space of the problem domain. In the process of solving a goal, there may be many branches that can be taken in an attempt to prove a goal. Rather than selecting a branch at random, intelligent behavior involves evaluating (by some criteria) each possible branch that may lead toward the solution of a goal to determine which branch is more likely to lead to a solution. In this evaluation, certain important **factors** are examined to weight various branches. For example, during a medical exam, a patient may complain of dizziness, nausea, fever, headache, and itchy feet. The doctor may know of thousands of possible diseases, conditions, allergies, etc. To narrow the search, the doctor will try to find a pathology that accounts for these symptoms. There may be some diseases that account for all 5 symptoms, others that might account for 4 out of the 5 symptoms, and so on. In this manner, the practitioner sorts and prunes his list of possible pathologies. Competency evaluation will be based on how likely an agent's branch will be successful (based on a weighted factor analysis) and how likely the collaborator's branch will be successful (based on a weighted factor analysis and a probabilistic model of the collaborator's knowledge).

In Section 3 we will sketch out how this calculation is made, present several mode selection schemes based on this factor analysis, and show the results of analytical evaluation of these schemes. In Section 4 we will present the methodology and results of using these schemes in a simulated dialogue environment.

3 Mathematical Analysis of Efficiency

Our model of best-first search assumes that for each goal there exists a set of n factors, $f_1, \ldots, f_n$, which are used to guide the search through the problem-solving space. Associated with each factor are two weights, w_i, which is the percentage of times a successful branch will have that factor and x_i which is the percentage of all branches that satisfy f_i. If an agent, a, knows $q_1^a, \ldots, q_n^a$ percentage of the knowledge concerning factors $f_1, \ldots, f_n$, respectively, and assuming independence of factors, using Bayes' rule an agent can calculate the success likelihood of each

[1] Since each participant is carrying out initiative evaluation independently, there may be conflicts on who should be in control. Numerous researchers have studied how negotiation may be used to resolve these conflicts (Guinn, 1994; Guinn, 1993a; Lambert and Carberry, 1992; McRoy, 1993; Sidner, 1993)

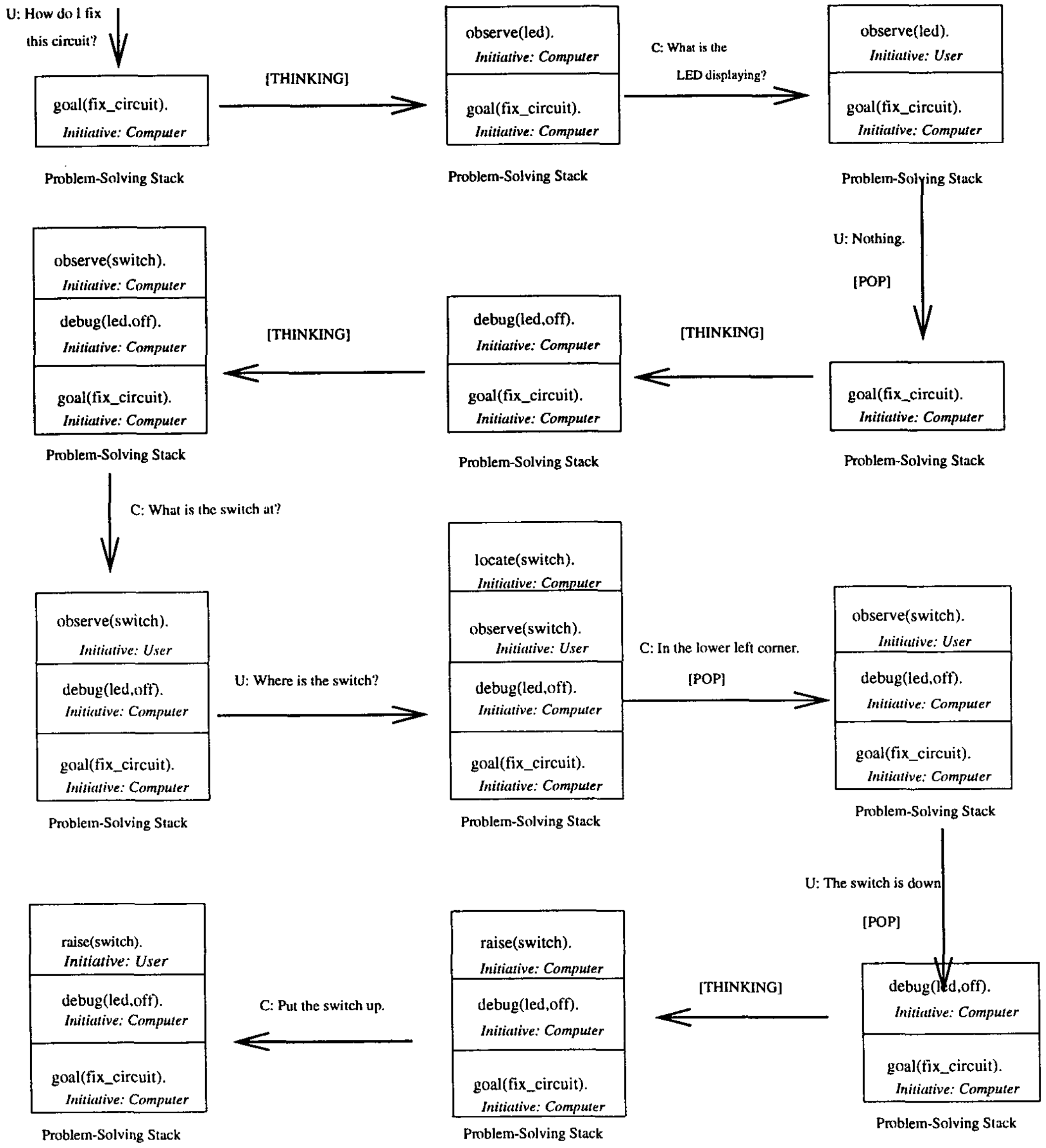

Figure 1: Tracking Initiative via Explicit Initiative-Changing Utterances and Problem-Solving Stack Manipulation

possible branch of a goal G that it knows:

$$p(b) = 1 - \prod_{i=1}^{n} 1 - F(i) w_i \frac{(1/k)}{x_i} \qquad (1)$$

where b is a branch out of a list of k branches and $F(i) = 1$ if the agent knows branch b satisfies factor f_i and $F(i) = x_i(1 - q_i^a)$ otherwise. [Note: $x_i(1 - q_i^a)$ is the probability that the branch satisfies factor f_i but the agent does not know this fact.] We define the sorted list of branches for a goal G that an agent knows, $[b_1^a, \ldots, b_k^a]$, where for each b_i^a, $p(b_i^a)$ is the likelihood that branch b_i^a will result in success where $p(b_i^a) >= p(b_j^a), \forall i < j$.

3.1 Efficiency Analysis of Dialogue Initiative

For efficient initiative-setting, it is also necessary to establish the likelihood of success for one's collaborator's 1^{st}-ranked branch, 2^{nd}-ranked branch, and so on. This calculation is difficult because the agent does not have direct access to its collaborator's knowledge. Again, we will rely on a probabilistic analysis. Assume that the agent does not know exactly what is in the collaborator's knowledge but does know the *degree* to which the collaborator knows about the factors related to a goal. Thus, in the medical domain, the agent may know that the collaborator knows more about diseases that account for dizziness and nausea, less about diseases that cause fever and headache, and nothing about diseases that cause itchy feet. For computational purposes these degrees of knowledge for each factor can be quantified: the agent, a, may know percentage q^a of the knowledge about diseases that cause dizziness while the collaborator, c, knows percentage q^c of the knowledge about these diseases. Suppose the agent has 1) a user model that states that the collaborator knows percentages $q_1^c, q_2^c, \ldots, q_n^c$ about factors $f_1, f_2, \ldots, f_n$, respectively and 2) a model of the domain which states the approximate number of branches, $\mathcal{N}$. Assuming independence, the expected number of branches which satisfy all n factors is $ExpAllN = \mathcal{N} \prod_{i=1}^{n} x_i$. Given that a branch satisfies all n factors, the likelihood that the collaborator will know that branch is $\prod_{i=1}^{n} q_i^c$. Therefore, the expected number of branches for which the collaborator knows all n factors is $ExpAllN \prod_{i=1}^{n} q_i^c$. The probability that one of these branches is a success-producing branch is $1 - \prod_{i=1}^{n} 1 - w_i \frac{(1/k)}{x_i}$ (from Equation 1). By computing similar probabilities for each combination of factors, the agent can compute the likelihood that the collaborator's first branch will be a successful branch, and so on. A more detailed account of this evaluation is given by Guinn (1993b; 1994).

We have investigated four initiative-setting schemes using this analysis. These schemes do not necessarily correspond to any observable human-human or human-computer dialogue behavior. Rather, they provide a means for exploring proposed dialogue initiative schemes.

Random In Random mode, one agent is given initiative at random in the event of a conflict. This scheme provides a baseline for initiative setting algorithms. Hopefully, a proposed algorithm will do better than Random.

SingleSelection In SingleSelection mode, the more knowledgeable agent (defined by which agent has the greater total percentage of knowledge) is given initiative. The initiative is set throughout the dialogue. Once a leader is chosen, the participants act in a master-slave fashion.

Continuous In Continuous mode, the more knowledgeable agent (defined by which agent's first-ranked branch is more likely to succeed) is initially given initiative. If that branch fails, this agent's second-ranked branch is compared to the other agent's first-ranked branch with the winner gaining initiative. In general if Agent 1 is working on its i^{th}-ranked branch and Agent 2 is working on its j^{th}-ranked branch, we compare $p^{A_1}(b_i^{A_1})$ to $p^{A_2}(b_j^{A_2})$.

Oracle In Oracle mode, an all-knowing mediator selects the agent that has the correct branch ranked highest in its list of branches. This scheme is an upper bound on the effectiveness of initiative setting schemes. No initiative setting algorithm can do better.

As knowledge is varied between participants we see some significant differences between the various strategies. Figure 2 summarizes this analysis. The x and y axis represent the amount of knowledge that each agent is given[2], and the z axis represents the percentage of branches explored from a single goal. SingleSelection and Continuous modes perform significantly better than Random mode. On average Continuous mode results in 40% less branches searched per goal than Random. Continuous mode

[2] This distribution is normalized to insure that all the knowledge is distributed between each agent. Agent 1 will have $q_1 + (1 - q_1)(1 - q_2)\frac{q_1}{q_1 + q_2}$ percent of the knowledge while Agent 2 will have $q_2 + (1 - q_1)(1 - q_2)\frac{q_2}{q_1 + q_2}$ percent of the knowledge. If $q_1 + q_2 = 0$, then set $q_1 = q_2 = 0.5$.

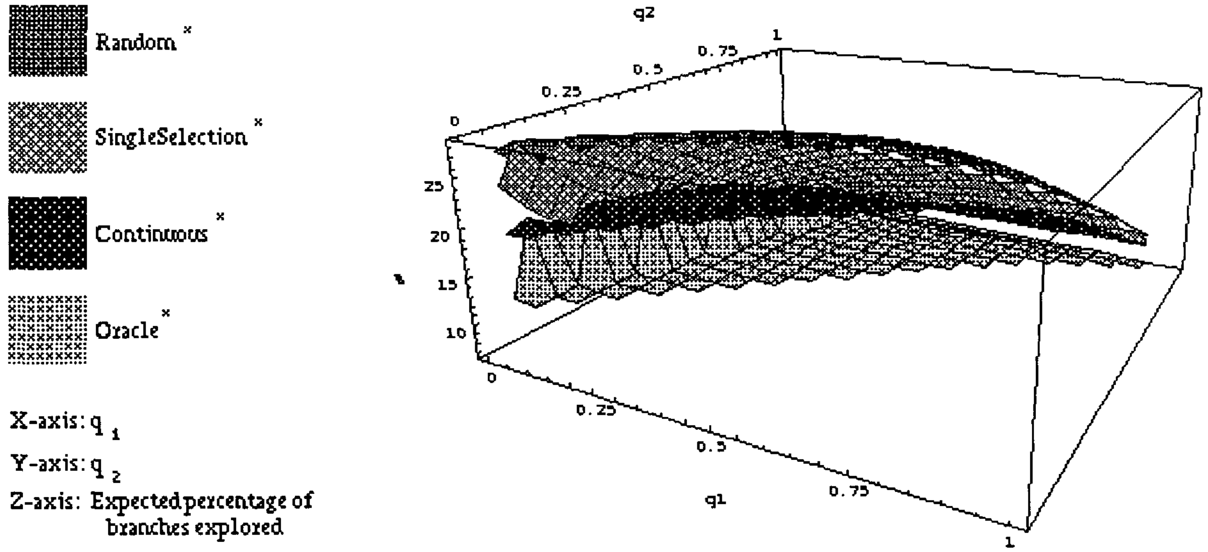

Figure 2: An Analytical Comparison of Dialogue Initiative-Setting Schemes

performs between 15-20% better than SingleSelection. The large gap between Oracle and Continuous is due to the fact that Continuous initiative selection is only using limited probabilistic information about the knowledge of each agent.

4 Computer Simulations

The dialogue model outlined in this paper has been implemented, and computer–computer dialogues have been carried out to evaluate the model and judge the effectiveness of various dialogue initiative schemes. In a methodology similar to that used by Power (1979), Carletta (1992) and Walker (1993), knowledge is distributed by a random process between agents, and the resulting interaction between these collaborating agents is observed. This methodology allows investigators to test different aspects of a dialogue theory. Details of this experimental strategy are given by Guinn (1995).

4.1 The Usage of Computer-Computer Dialogues

The use of computer-computer simulations to study and build human-computer dialogue systems is controversial. Since we are building computational models of dialogue, it is perfectly reasonable to explore these computational models through computer-computer simulations. The difficulty lies in what these simulations say about human-computer or computer-computer dialogues. This author argues that computer-computer simulations are one layer in the multi-layer process of building human-computer dialogue systems. Computer-computer simulations allow us to evaluate our computational models and explore issues that can not be resolved analytically. These simulations can help us prune out some mechanisms and suggest mechanisms that may work well in a human-computer system. For instance, if the dialogues that result in a computer-computer conversation are incoherent to a human observer, this suggests that the dialogue mechanisms employed may be inappropriate for a human-computer system. On the other hand, if the resulting dialogues are coherent and exhibit features that are desired in a human-computer system, this suggests that these mechanisms may work well in a human-computer system. The final test, of course, must be in the implementation of a human-computer dialogue system. The scarcity of such systems suggests that it is an extremely expensive process to build a functional human-computer dialogue system, and computer-computer simulations can assist in reducing these costs.

4.2 Simulation Results

A murder mystery domain was created with 22 suspects. An annotated example dialogue from a computer–computer collaboration in this domain is presented in Figure 3. Agents were given partial information through a random process. By using either Random mode or Continuous mode, we can evaluate the effect of those mechanisms in this experimental environment. Table 2 presents results computed from 5008 collaborations where the agents

Speaker	Utterance
Watson	Who is the murderer of Lord Dunsmore?[a]
Holmes	Is it the case that Suspect16 is the murderer of Lord Dunsmore?
Watson	Is it the case that Suspect16 had a motive to murder Lord Dunsmore?
Holmes	Suspect16 had a motive to murder Lord Dunsmore.
Watson	Is it the case that Suspect16 had access to the poison?
Holmes	I am unable to determine whether Suspect16 had access to the poison.
Watson	I am unable to determine whether Suspect16 is the murderer of Lord Dunsmore.
Holmes	I am unable to determine whether Suspect16 is the murderer of Lord Dunsmore.
Holmes	Who is the murderer of Lord Dunsmore?[b]
Watson	Is it the case that Suspect10 is the murderer of Lord Dunsmore?
Holmes	Is it the case that Suspect10 had a motive to murder Lord Dunsmore?
Watson	Suspect10 had a motive to murder Lord Dunsmore.
Holmes	Is it the case that Suspect10 had access to the poison?
Watson	Suspect10 had access to the poison.
Holmes	Is it the case that Suspect10 had an opportunity to administer the poison?
Watson	I am unable to determine whether Suspect10 had an opportunity to administer the poison.
Holmes	I am unable to determine whether Suspect10 is the murderer of Lord Dunsmore.
Watson	I am unable to determine whether Suspect10 is the murderer of Lord Dunsmore.
Watson	Is it the case that Suspect9 is the murderer of Lord Dunsmore?
Holmes	Is it the case that Suspect7 is the murderer of Lord Dunsmore?[c]
Watson	I have proven that Suspect9 has a motive to murder Lord Dunsmore and Suspect9 had access to the poison.[d]
Holmes	I have proven that Suspect7 had access to the poison, Suspect7 had an opportunity to administer the poison, and Suspect7 has a criminal disposition.[e]
Watson	Suspect7 is the murderer of Lord Dunsmore.[f]

[a] Watson gives control of the investigation over to Holmes. Each participant uses the Continuous Mode algorithm to determine who should be in control.

[b] Holmes is giving up control of directing the investigation here.

[c] Holmes is challenging Watson's investigative choice.

[d] Watson negotiates for his choice.

[e] Holmes negotiates for his choice.

[f] Watson now has enough information to prove that Suspect7 is the murderer.

Figure 3: A Sample Dialogue

had to communicate to solve the task.

	Random	Continuous
Times (secs)	82.398	44.528
# of Utterances	39.921	26.650
Suspects Examined	6.188	3.412

Table 2: Data on 5008 Non-trivial Dialogues from the Murder Mystery Domain

5 Extension to Human–Computer Dialogues

Currently, two spoken-dialogue human-computer systems are being developed using the underlying algorithms described in this paper. The Duke Programming Tutor instructs introductory computer science students how to write simple Pascal programs by providing multiple modes of input and output (voice/text/graphics) (Bierman et al., 1996). The Advanced Maintenance Assistant and Trainer (AMAT) currently being developed by Research Triangle Institute for the U.S. Army allows a maintenance trainee to converse with a computer assistant in the diagnosis and repair of a *virtual* M1A1 tank. While still in prototype development, preliminary results suggest that the algorithms that were successful for efficient computer–computer collaboration are capable of participating in *coherent* human–machine interaction. Extensive testing remains to be done to determine the actual gains in efficiency due to various mechanisms.

One tenet of our theory is that proper initiative setting requires an effective user model. There are several mechanisms we are exploring in acquiring the kind of user model information necessary for the previously described dialogue mode algorithms. Stereotypes (Rich, 1979; Chin, 1989) are a valuable tool in domains where user classification is possible and relevant. For instance, in the domain of military equipment maintenance, users can be easily classified by rank, years of experience, equipment familiarity and so on. An additional source of user model information can be dynamically obtained in environments where the user interacts for an extended period of time. A tutoring/training system has the advantage of knowing exactly what lessons a student has taken and how well the student did on individual lessons and questions. Dynamically modifying the user model based on on-going problem solving is difficult. One mechanism that may prove particularly effective is negotiating problem-solving strategies (Guinn, 1994). The quality of a collaborator's negotiation reflects the quality of its underlying knowledge. There is a tradeoff in that negotiation is expensive, both in terms of time and computational complexity. Thus, a synthesis of user modeling techniques will probably be required for effective and efficient collaboration.

6 Acknowledgements

Work on this project has been supported by grants from the National Science Foundation (NSF-IRI-92-21842), the Office of Naval Research (N00014-94-1-0938), and ACT II funding from STRICOM for the Combat Service Support Battlelab.

References

A. Bierman, C. Guinn, M. Fulkerson, G. Keim, Z. Liang, D Melamed, and K Rajagopalan. 1996. Goal-Oriented multimedia dialogue with variable initiative. In *submitted for publication.*

D.G. Bobrow, 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.

J. Carletta. 1992. Planning to fail, not failing to plan: Risk-taking and recovery in task-oriented dialogue. In *Proceedings of the 14th International Conference on Computational Linguistics (COLING-92)*, pages 896–900, Nantes, France.

D.N. Chin. 1989. KNOME: Modeling what the user knows in UC. In A. Kobsa and W. Wahlster, editors, *User Models in Dialog Systems*, pages 74–107. Springer-Verlag, New York.

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

C.I. Guinn. 1993a. Conflict resolution in collaborative discourse. In *Computational Models of Conflict Management in Cooperative Problem Solving, Workshop Proceedings from the 13th International Joint Conference on Artificial Intelligence*, Chambery, France, August.

Curry I. Guinn. 1993b. A computational model of dialogue initiative in collaborative discourse. *Human-Computer Collaboration: Reconciling Theory, Synthesizing Practice, Papers from the 1993 Fall Symposium Series, Technical Report FS-93-05.*

Curry I. Guinn. 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.

Curry I. Guinn. 1995. The role of computer–computer dialogues in human–computer dialogue system development. *AAAI Spring Symposium on Empirical Methods in Discourse Interpretation and Generation, Technical Report SS-95-06.*

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

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

S. McRoy. 1993. Misunderstanding and the negotiation of meaning. *Human-Computer Collaboration: Reconciling Theory, Synthesizing Practice, Papers from the 1993 Fall Symposium Series, AAAI Technical Report FS-93-05*, September.

R. Power. 1979. The organization of purposeful dialogues. *Linguistics*, 17.

E. Rich. 1979. User modeling via stereotypes. *Cognitive Science*, 3:329–354.

C. L. Sidner. 1993. The role of negotiation in collaborative activity. *Human-Computer Collaboration: Reconciling Theory, Synthesizing Practice, Papers from the 1993 Fall Symposium Series, AAAI Technical Report FS-93-05*, September.

R.W. Smith and D.R. Hipp. 1995. *Spoken Natural Language Dialog Systems: A Practical Approach.* Oxford University Press, New York.

R.W. Smith, D.R. Hipp, and A.W Biermann. 1992. A dialog control algorithm and its performance. In *Proceedings of the 3rd Conference on Applied Natural Language Processing.*

M. Walker and S Whittaker. 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.

M. A. Walker. 1993. *Informational Redundancy and Resource Bounds in Dialogue.* Ph.D. thesis, University of Pennsylvania.

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

S.J. Young 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.

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