MECHANISMS FOR MIXED-INITIATIVE

HUMAN–COMPUTER COLLABORATIVE DISCOURSE

Curry I. Guinn
Department of Computer Science
Duke University
cig@cs.duke.edu

Abstract: In this paper, we examine three inter-related efficiency-improving dialogue behaviors: automatic dialogue initiative setting, negotiation for conflict resolution and summaries for plan recognition assistance. We show how to incorporate these behaviors in a task-oriented human–computer dialogue system, and we evaluate these mechanisms both analytically and via computer–computer dialogue simulation.

Subjects: Dialogue, mixed-initiative, negotiation, summaries.

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. In the event of a conflict over which agent should handle the current task, the participants must efficiently negotiate the next move and proceed. To insure that the collaborators detect conflicts (which requires proper plan recognition), agents must announce the results of certain task processing. Specifically in this paper, we examine three efficiency improving dialogue behaviors, their incorporation into an automatic dialogue system, and their performance improving capabilities:
1. **Automatic setting of initiative levels.** 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 following dialogue fragment occurs between a human user and a computer collaborator.

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

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.

2. **Automatic negotiation for conflict resolution.** Our model assumes that each agent has incomplete information about the domain and an incomplete user model of its collaborating agent. Thus the agents may not agree on who should be in control. One mechanism for dealing with this conflict is negotiation. In the following dialogue, there is a conflict about which path to take in fixing a car.

   Each agent gives evidence to support its choice over its collaborator’s.

   [Two people trying to find out why a car won’t start]

   1. **Jordan:** Help me get the distributor cap off so we can check the spark plugs.
   2. **Chris:** The lights were probably left on last night.
   3. **Jordan:** It’s the battery.
   4. **Chris:** The voltage on the battery is fine.

We will present a theory of when such negotiations occur and how the content of these negotiation utterances is generated.
3. **Automatic generation of non-obligatory task summaries.** Proper initiative-setting mechanisms and efficient negotiation require proper plan recognition. In our model an agent who is passively following the lead of its collaborator should become less passive if the collaborator chooses paths that prove unsuccessful. However, in a domain where an agent has limited knowledge, it may be difficult for an agent to determine that its collaborator’s plan is proving unsuccessful. Thus we explore the use of non-obligatory goal processing announcements\(^1\) that will assist in proper plan recognition.

## 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 \textit{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 \textit{level of initiative in the dialogue should mirror the level of initiative in the task} (which is a corollary to Grosz’s (1978) \textit{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 \textbf{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.

---

\(^1\)In our model, if an agent solves or fails to solve a goal that has been requested by its collaborator, the agent is \textit{obliged} to announce this result. A non-obligatory announcement is the summarizing of an attempt to prove a goal that has not been explicitly requested by the collaborator.
2.1. The Setting of Initiative Levels

In all previous dialogue systems, the initiative levels for each goal are defined \textit{a priori} before the dialogue begins. 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.

1) 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. In Figure 1, all initiative changes in the dialogue are explained by explicit initiative-changing utterances or by popping of the problem-solving stack due to goal resolution.

2) 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 5 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 6 we will
present the methodology and results of using these schemes in a simulated dialogue environment.

3. AUTOMATING NEGOTIATION FOR CONFLICT RESOLUTION

In practice, reliable initiative-setting mechanisms may be difficult to achieve because these mechanisms require accurate user model information and accurate domain knowledge. Thus, total reliance on initiative mechanisms to reduce conflicts and produce more efficient collaborative problem-solving is unsatisfactory. Negotiation is a common mechanism for resolving conflicts in human–human interaction. While there has been a great deal of research on computational models of human–computer negotiation (Lambert and Carberry, 1992; McRoy, 1993; Sidner, 1993), this research has emphasized negotiation to resolve user (or computer) misconceptions. It has not emphasized how negotiation may produce more efficient collaborative problem-solving between humans and computers by pruning the “collaborative” search space.

In our model of negotiation, when an agent detects a conflict with its collaborator — the agent believes that branch $b_a$ is more likely to succeed for solving goal $G$ while its collaborator believes branch $b_c$ will be more successful — that agent presents the factor information it knows for branch $b_a$. This factor information, which the collaborator may not have known, can now be used by the collaborator in its evaluation of the possible branches. The goal of the negotiating agent is to raise the collaborator’s evaluation of branch $b_a$, thus making the collaborator more likely to assist in the agent’s plan.²

In Section 5 we will sketch out two negotiation strategies and present the results of analytical evaluation of these strategies. In Section 6 the methodology and results of using these strategies in a simulated dialogue environment will be presented.

4. AUTOMATING THE GENERATION OF NON-OBLIGATORY SUMMARIES

In our model of dialogue, certain utterances are obligatory. For instance, when an agent satisfies (or fails to satisfy) a goal and its collaborator has previously requested the result of that goal, the agent is obliged to announce it. Is it ever beneficial to announce goal results that haven’t been requested?

²Note that there could be negative factor information as well. For instance, the agent might present evidence that branch $b_c$ is less likely to be a success-producing branch. Our model currently does not utilize negative factors.
Notice that initiative-setting mechanisms and negotiation strategies require the ability to detect differences between an agent’s plan and that of its collaborator. These mechanisms are rendered almost useless without this ability. However, plan recognition is one of the more difficult problems facing intelligent agents. In domains where the agent has limited knowledge, mistakes in plan recognition will be common. Utterances that explicitly outline the plan an agent is taking can assist greatly in this process. Our research focuses on the announcement of certain goal failures. Suppose an agent insists on pursuing branch $b_a$ in solving goal $G$ even though it knows its collaborator wishes to pursue an alternative branch. Now the collaborator will be passive in solving goal $G$. However, if branch $b_a$ proves to be unsuccessful, we want the collaborator to have the option of becoming more assertive again. However, unless the agent notifies its collaborator that branch $b_a$ has failed, the collaborator may not realize that the agent is pursuing different branches. In the analysis in the following sections, we will look at the possible advantages and disadvantages that these non-obligatory summaries may have on initiative setting schemes and negotiation strategies.

5. 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 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 possible branch of a goal $G$ that it knows:

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

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 to be $[b_1^a, \ldots, b_k^a]$ and $p(b_i^a)$ to be the likelihood that branch $b_i^a$ will result in success where $p(b_i^a) \geq p(b_j^a), \forall i < j$. 

7
5.1. Efficiency Analysis of Dialogue Initiative

For efficient initiative-setting, it is also necessary to establish the likelihood of success for one’s collaborator’s 1st-ranked branch, 2nd-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 may know percentage $q^a$ of the knowledge about diseases that cause dizziness while the collaborator 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 $q^c_1, q^c_2, \ldots, q^c_n$ about factors $f_1, f_2, \ldots, f_n$, respectively and 2) a model of the domain which states the approximate number of branches, $N$. Assuming independence, the expected number of branches which satisfy all $n$ factors is $\text{ExpAll} = N \prod_{i=1}^n x_i$. Given that a branch satisfies all $n$ factors, the likelihood that the collaborator will know that branch is $\prod_{i=1}^n q^c_i$. Therefore, the expected number of branches for which the collaborator knows all $n$ factors is $\text{ExpAll} \prod_{i=1}^n q^c_i$. The probability that one of these branches is a success-producing branch is $1 - \prod_{i=1}^n 1 - w_i \frac{1}{x_i}$ (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 Anonymous (1993; 1994).

We have investigated four initiative-setting schemes using this analysis:

Random In Random mode, one agent is given initiative at random in the event of a conflict.

SingleSelection In SingleSelection mode, the more knowledgeable agent (defined by which agent has the greater total percentage of knowledge) is given initiative.

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 $p_{A1}(b_{i}^{A1})$ to $p_{A2}(b_{j}^{A2})$.

**Oracle**  In Oracle mode, an all-knowing mediator selects the agent that has the correct branch ranked
highest in its list of branches.

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\(^3\), 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 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.

\[Q = \frac{q_{1} + (1-q_{1}) \cdot \frac{q_{1}}{q_{1} + q_{2}}}{q_{1} + q_{2}}\]

If $q_{1} + q_{2} = 0$, then set $q_{1} = q_{2} = 0.5$.

---

\(^3\)This distribution is normalized to insure that all the knowledge is distributed between each agent. Agent 1 will have
$q_{1} + (1-q_{1}) \cdot \frac{q_{1}}{q_{1} + q_{2}}$ percent of the knowledge while Agent 2 will have $q_{2} + (1-q_{2}) \cdot \frac{q_{2}}{q_{1} + q_{2}}$ percent of the knowledge.

**Figure 2: An Analytical Comparison of Dialogue Initiative-Setting Schemes**
5.2. Efficiency Analysis of Negotiation

Using a similar probabilistic reasoning, we can calculate the effects of various negotiation schemes. In one scheme, OneNegotiation, the agents negotiate once if a conflict ensues in the solving of a goal. The winner of the negotiation takes the initiative. In a second scheme, AllNegotiation, the agents negotiate if a conflict ensues in the solving of a goal. The winner will initially take the initiative. However, if the winner's branch proves unsuccessful, the agents negotiate again. Figure 3 illustrates the relative merits of OneNegotiation over Random and AllNegotiation over OneNegotiation.

![Figure 3: An Analytical Comparison of Negotiation Strategies](image)

5.3. Efficiency Analysis of Non-Obligatory Summaries

In the worse case, an agent may be incapable of determining what branch its collaborator is working on. In the AllNegotiation strategy, an agent would not be capable of determining when the controlling agent has abandoned its 1st-ranked branch. Thus the AllNegotiation scheme would degenerate to perform no better than the OneNegotiation scheme. In the case of Continuous mode, each agent needs to know where its collaborator is working on its ith-ranked branch in order to effectively calculate who should be in control. Thus the Continuous strategy may degenerate into SingleSelection without proper plan recognition.

6. 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, negotiation strategies, and the use of non-obligatory summaries. 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 Anonymous (1995).

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 4. Agents were given partial information through a random process. By turning on and off various dialogue mechanisms, we can evaluate the effect of those mechanisms in this experimental environment. Table 1 presents results computed from 5008 collaborations where the agents had to communicate to solve the task. This data gives us empirical validation of the effects of applying certain dialogue mechanisms in our underlying model. For instance, we can see that both the Continuous initiative-setting algorithm and the use of negotiation can substantially increase the efficiency of the joint problem-solving. While Non-Obligatory Summaries did not significantly affect the Continuous initiative-setting algorithm, they did substantially increase the efficacy of negotiation.

| Summaries | Yes | Yes | Yes | Yes | Yes | Yes | Yes | Yes |
|---|---|---|---|---|---|---|---|---|
| Negotiation | No | Yes | No | Yes | No | Yes | No | Yes |
| Continuous Mode | No | No | Yes | Yes | No | No | Yes | Yes |
| Time (secs.) | 31.822 | 46.685 | 41.619 |
| Utterances | 45.035 | 25.276 | 29.415 |
| Suspects | 6.135 | 6.655 |
| Examined | 3.400 | 3.289 |

Table 1: Data on 5008 Non-trivial Dialogues from the Murder Mystery Domain

REFERENCES

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.

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

C.I. Guinn. 1994. *Meta-Dialo gue 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.

C.I. Guinn. 1995. Computer–computer dialogues in human-computer dialogue system development. *AAAI Spring Symposium on Empirical Methods in Discourse Interpretation and Generation*, March. To appear.
| Speaker | Utterance |
|---------|-----------|
| Watson  | Who is the murderer of Lord Dunsmore?² |
| 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.¹ |
| Holmes  | Who is the murderer of Lord Dunsmore?³ |
| 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.⁴ |
| 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?⁵ |
| Watson  | I have proven that Suspect9 has a motive to murder Lord Dunsmore and Suspect9 had access to the poison.⁶ |
| 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.⁷ |
| Watson  | Suspect7 is the murderer of Lord Dunsmore.⁸ |

²Watson gives control of the investigation over to Holmes. Each participant uses the Continuous Mode algorithm to determine who should be in control.
³This statement is an example of a Non-Obligatory Summary. Note that Watson never directly asks whether Suspect16 is the murderer of Lord Dunsmore. Non-Obligatory Summaries may be required for proper plan recognition.
⁴Holmes is giving up control of directing the investigation here.
⁵Another Non-Obligatory Summary.
⁶Holmes is challenging Watson’s investigative choice.
⁷Watson negotiates for his choice.
⁸Holmes negotiates for his choice.
⁹Watson now has enough information to prove that Suspect7 is the murderer.

Figure 4: A Sample Dialogue
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.

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 Biemann. 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.