Initiatory and Reactive System Roles

in Human Computer Discourse

Kevin Matthews

CUCS-151-85

Kevin Matthews

Department of Computer Science
Columbia University
New York, New York 10027

Abstract

User interfaces for a variety of computer systems will need to develop more conversational abilities in order to communicate effectively and cooperatively with the user. In particular, such systems need to be able to shift flexibly from asking questions and making requests of the user to answering similar questions and requests from the user. We provide an analysis of the discourse-level, linguistic phenomena involved in such shifts. We further specify the information such an automated system would need to maintain to support this linguistic capability and how it would use the information. We conclude by showing output from an automated student advising program ADVISOR exemplifying a specific case of taking the initiator role to provide unrequested information.

[1]This work was partially supported by ONR grant N00014-82-k-0256 and by AT&T Bell Laboratories, 600 Mountain Ave., Murray Hill, NJ 07974

Table of Contents

1 Introduction

Our guiding concern in the research described in this paper is to contribute to the development of better natural language interfaces for a wide range of computer systems including data base, CAI, help, information retrieval, and expert systems. Those human/machine interactions which are likely to be the most helpful to the non-expert computer user are those that most resemble human/human interactions in comparable domains. Human/human problem-solving dialogues are characterized by a more flexible, give-and-take interplay between the participants than is true of the typical examples of these computer systems. In particular, it is seldom the case in natural dialogue that one conversant asks all the questions while the other provides all the answers. In the standard interface for database retrieval systems the user asks all the questions while for an expert system the system itself poses all the questions. In this paper we consider what would be necessary to enable a conversant, such as an automated system, to shift appropriately between responding to the user and directing the discussion during a sub-dialogue. We restrict our attention to problem-solving dialogue as the most relevant type for human/machine interaction.

We begin by establishing motivations for which an advising system would need to be able to shift from the more reactive responder's role in the conversation to the more active controlling role and vice versa.[2] These observations are drawn from protocols of human/human advising sessions.

We then describe the nature of the discourse-level linguistic roles of initiator and reactor and specify the information that an automated system would need to have at any point in the dialogue in order to manage these roles. Based on this information an algorithm is described to enable the advising system to choose among its alternatives in forming its next utterance. After a discussion of the alternatives appropriate to each role we consider in detail a particular occasion for an advising system to shift from the reactor role to the initiator along some special considerations that arise in problem-solving systems. For this subproblem we present the implementation of a partial solution. While we are not specifically concerned with cognitive validity, our ultimate goal is to develop a model that will support the managing of transitions in the initiator/reactor roles for a wide variety of the speaker's goals.

We have carried out a partial implementation of our approach in an experimental system

[2]It is important to note that the term "role" is being used here exclusively in a linguistic sense, not with regard to its social or psychological connotations.

ADVISOR which functions as a faculty advisor of undergraduate computer science majors for such problems as course selection. ADVISOR is structured as an enhanced question-answering system with a database retrieval capability and a mini-expert system capable of reasoning about the computer science major. To these components we are adding the conversational capabilities discussed in this paper. Our interest in the ADVISOR system is as a laboratory for testing and modifying the present partial solution along with solutions to other problems of natural language processing. ADVISOR is not intended to demonstrate the feasibility of an automated student advisor as an engineering project.

1.1 The Roles of Initiator and Reactor

The most fundamental distinction between conversants in dyadic conversation is that of speaker/listener. Based on our observations of transcripts of human/human advising sessions the speaker's choice during his turn is between two linguistic roles which we distinguish as follows:

1. **initiator**- controls the conversation during a segment of dialogue, by asking questions, requesting information, or by informing as stage-setting for either of these goals. The dominant expectation is that the other conversant in the dialogue will respond to the direction supplied by the initiator. In example 1 the student is the initiator from turn #36 until #43.
2. **reactor**- responds to the questions or requests for information from the initiator or makes back-channel responses that indicate his continuation of the reactor role without contributing content (e.g. "mmhmm").

It is important to note that these roles may change independently of turn and topic changes. In example 3 below, turn #46, for instance, the advisor changes his role from reactor to initiator while changing the topic from elective courses to required ones. Here the role and the topic both shift at the same time: from reactor to initiator and from the topic of elective courses to required courses. In example 2, line #4 however, the role shifts from reactor to initiator while the topic remains the same. In example 4 below, the topic shifts (at the points marked by "ok") while the speaker remains in the initiator role.

1.2 Motivation: Analysis of a Sample Dialogue

The present work is based on a study of transcripts of faculty members advising students about course selection. The goal of this study was to formulate a model of management of the initiator/reactor role sufficiently precise to provide a computational basis for a conversational computer program. The following transcript illustrates several of the issues involved.

36. s: I might be out of six credits is that what you are saying?	**;initiator**
37. p: That's right. Certainly if you do plan to take PLT I that will not count towards the master's.	**;responder**
38. s: Ok. Is II counted towards the master's?	**;initiator**
39. p: Yes. Yes.	**;responder**
40. s: ok	**;abandon**
41. p: these are the only two courses in question. PLT I, is without question doesn't count. The operating systems course is in question. So,	**;responder**
42. s: Is there a possibility of my transferring some credits from some Stevens courses that I took?	**;initiator**
43. p: Are they graduate level?	**;initiator**
44. s: yeah	**;responder**
45. p: Oh no. They, in general we don't transfer any credits from any university to Columbia unless it's uh, it's a uh, a rather uh, familiar undergraduate course. Graduate courses can never transfer.	**;responder**

Example 1, from the transcripts

In this example, taken from the transcript of a dialogue between a student and her faculty advisor, the student (s) has the initiator role in the beginning of the segment #36. She asks a question and the advisor (p) answers, confirming his role as reactor. The student's "ok" in line #38 and again in line #40 are markers of the type that Reichman [12] identifies as boundaries for discourse segments based on focus, which she calls context spaces. Here we claim that linguistic markers such as "ok" may have the additional function of marking the boundary of an assignment of roles. In #38 the "ok" indicates that the initiator (the student) has completed her goal as initiator. As a result, occupancy of the initiator role is open for bids, as it were. In particular, the reactor may attempt to gain the initiator role at this point. In the case of #38, however, the student makes a bid to continue the initiator role by asking another question. In the similar case of line #40 the student indicates completion of her initiator goal ("ok") and abandons the role by forbearing to ask another question. At #41 the advisor does not seize the initiator role forcefully with a goal of his own on a new topic. Rather he summarizes his previous response as a kind of temporizing or, one could even say, back-channel activity. As a result, the initiator role remains open for the student to retake with a minimum of linguistic effort in line #42. In #43 the advisor takes the initiator role to ask for additional information needed to answer the pending question of #42. The student responds in #44 and the advisor assumes the reactor role again in #45 to answer the pending question. The role changes in #42-45 occur without linguistic markers, but nevertheless happen smoothly with neither confusion nor the need for repair.

It is clear from this example that if an advising or problem-solving system is to cooperate in providing information in a natural fashion to the user it must be able at least to take the initiative to ask for more information when the user's question does not provide all that is needed. Other reasons for which the system would want to take the initiative include:

- seek clarification: to ask a question to resolve a pronoun reference, for example. Occurs in turn 4 of example 2 below.
- correct a misconception: system should correct misconception implied in user's question.
- provide unrequested information: consistent with the Gricean maxim of completeness, if the system has additional, pertinent information of which the user is not aware then it should inform the user.
- redirect discussion: to introduce a new topic or problem that the user should know about. Occurs in turn 46 of example 3 below.
- identify conflict between user's goal and system's: for instance, in advising domain, student may wish to pursue some action which the system knows would hurt the student's schedule.

In the data we studied, the most straightforward motivation for the system to take the reactor role is to answer a question from the user. Taking the reactor role may also occur when the system has completed the goal that motivated its taking the initiator role and abandons that role or when the system yields to an attempt by the user to take the initiator role.

1. s: um yes. I didn't take Fortran language.

2. p: umhm.

3. s: I only took Pascal. If I take that would I get a credit for it? It's a beginning course.

4. p: To take another 1000? **;;takes initiator role**

5. s: uh huh

6. p: no

7. s: no?

8. p: no

Example 2, from transcripts
Taking Initiative for Clarification

1.3 Related Work

Grosz [5] correlated the shift in focus in a problem-solving dialogue with the inherent structure of the underlying task. The task she studied, assembly of a mechanical device, is characterized by a strongly hierarchical organization. Each subtask of the assembly project is refelcted in the dialogue by a *focus space* which aids in determining which portions of the database may currently be relevant for such purposes as resolving definite pronoun reference. The domain of student advising, however, is much less structured with the result that dialogue in this domain tends to change topic freely. Grosz does not consider how a participant may change the topic if he or she wishes to nor does she address the issues of roles.

Reichman [12] carried Grosz's treatment of focus shift further. She identifies an organization of dialogue into segments according to focus which she called *context spaces.* Transitions between them are indicated by the use of linguistic markers (e.g. "ok", "so", etc.). Reichman further proposes that the relationships between context spaces, primarily according to content and rhetorical function, restrict their

---

43. s: ok um, how about computer arts-video games which one do you think will be better?

44. p: Well it depends on what you are interested in. Computer arts-video games actually it's an introduction to computer graphics.

45. s: umhm

46. p: And if you are interested in computer graphics this is your chance to learn about it. Uh, if you are interested in computer networks you should take the other one.
[pause]
Let me first check up. Do you have all your prerequisites? Let's go through the list: You have calc I,... **;initiates**

Example 3, from the transcripts

Taking Initiative to Redirect Discussion

---

possible orderings. These restrictions enable her to suggest an ATN-type grammar to parse or generate discourse which is well-formed as to topic shift. Her point of view in generation is, however, very different from ours. She considers the dialogue from the point of view of a detached third party. Her ATN generates a whole dialogue consisting of both participant's parts much as the more familiar ATN grammars generate a sentence. She does not consider generation from the point of view of a single conversant and the information such a participant would have at any given point and how he could use it to make his choices.

A number of researchers have developed experimental systems capable of some degree of mixed initiative. Pazzani's KNOBS [9] system can take the initiative for domain related purposes, such as an underspecified question. It accomplishes this by using a script or frame as the basis for constructing its database queries. If a slot in that frame is not filled, KNOBS can generate a question to fill it. As a result of its dependence on a frame the system can not generate a question in a similar way for communication purposes, such as clarification.

19. p: um alright then I would... Let me tell you a little bit about, about the way that the, the degree programs work for the School of General Studies and the Engineering School, that is, describe what courses go into a degree and then tell you what I recommend you start doing. Um, for everybody the first two years is pretty much, pretty much the same. It doesn't really, you know it is not particularized to computer science. So, the only computer science course really that anybody takes as a freshman or a sophomore is is the introductory computer programming course. And sometimes they take discrete mathematics course. Ok? Now after that, um that starts in the junior that people start taking a lot of computer science courses and the first semester the junior year they take um, blank sorry... data structures course ok, and if they haven't taken the discrete mathematics course already then they take it then. Ok? The second semester

Example 4, from the transcripts

Topic changes while role does not

Codd's RENDEZVOUS system [4] does have the ability to take the initiative for questions for communication purposes, but it does not have a way to decide when to refrain from doing so. It has a tendency therefore to dominate the interaction by asking numerous clarification questions to the point of frustrating the user.

Our goal is to provide a basis for a mixed-initiative system that overcomes the limitations of KNOBS and RENDEZVOUS. We intend to build on the work of Grosz and Reichman to permit management not only of topic shift, but also of the initiator and reactor roles. In particular, we find that Reichman's analysis of the role of linguistic markers suggests that additional use of linguistic markers may be made by conversants for managing roles. We note that some of the particle words that she identifies as indicators of a shift in topic cannot always be identified with that function in the data. In example 1, for instance, the use of the particle "ok" in lines #38 and #40 can be mapped neither to a substantive use since the speaker is not agreeing with a proposition in the previous utterance nor with a shift in topic since none occurs. We regard both occurrences as indicators rather of a shift in role. In each case the student s is indicating abandonment of the initiator role after having completed a goal for which the initiator role was needed. This is true despite the fact that in line #38 s immediately seizes the initiator role again.

To this extent, then the data supports the view that the initiator and reactor roles are finite, discourse-level linguistic resources which the conversants cooperate in managing so that the conversation may proceed in a orderly fashion and each participant may accomplish her goals.

2 A Partial Solution: Managing the Roles

We first consider what information will be necessary to support shifts in the initiator/reactor roles and then how this information can be used to make those shifts.

The information the system will use to carry on the dialogue consists of:

1. Its set of utterance-level goals (plans), which include domain-independent communication goals and domain-dependent goals.
2. Its set of domain-dependent session-goals, which are, in general, more elaborate sequences of subgoals.
3. Its current discourse position.

A system will need to take over the initiative in a dialogue only when it has some purpose to accomplish in doing so. We propose two kinds of system goals, each of which is represented in a plan formalism. The first type is a local, *utterance-level* plan whose function is to examine the current input utterance from the user and determine how to respond to it, whether by answering it directly or addressing some problem raised of which the user may or may not be aware. For example, if the utterance includes a pronoun whose referent cannot be determined by the system then the system would generate a question to ask the user to resolve it. Domain-independent, communication goals, as well as some domain-level goals, would be addressed by utterance-level plans. If the user's question were underspecified the system would ask for more information. The most preferred or expected response, that of answering the question, would also be represented as an utterance-level plan, to answer, that might consist of generating and executing a transaction against the database.

In addition, the system needs to have domain-dependent, *session-level* goals, also represented as plans. These do not depend on the content of any of the user's utterances to be activated. Rather, they are domain-level actions which the system is committed to address if the user does not address them first. For instance, in the student advising domain the system might have a session-level goal that the student have a schedule of courses to take for the following semester. If the student does not present a schedule then the system will be prepared to take the initiator role and introduce the problem. Another example in the same domain is to determine how many of the required courses the student has taken. This goal motivates the initiative-taking in example 3 turn #46. As we can see from this example a session goal may consist of a series of sub-goals. Activation of a session-level plan requires that the system have the initiator role. As reactor therefore, the system's session-goal slot would always be empty.

The two sets of system plans, utterance-level and session-level, are procedural kinds of knowledge which the system has already built-in. In addition, the system needs to have a dynamic model of what is happening in the dialogue at any moment. This information we call the system's *discourse position* and is represented in a data structure consisting of four attribute/value pairs. The implicit assumption is that the other conversant maintains a similar representation of his or her position. It may also be desirable for the system to maintain a version of the discourse structure representing the other conversant's presumed discourse position, although we have not implemented this option. What then constitutes the system's discourse position?

First, the system must know whether its current role is as initiator or reactor. Its options and the relative difficulty of exercising them will depend on its current role. If, for instance, the user's last utterance is ambiguous and the system wants to ask a clarifying question, it may have to use more forceful linguistic means to do so if it is the reactor than if it is the initiator.

Secondly, it is also clear that the system will need to know which is the current topic under discussion. Furthermore we have seen from our examples that topic shifts and role shifts are independent of each other and therefore should be represented separately. Therefore, the information that describes the conversant's current state in the discourse position must include both the current topic and the current role of the conversant in question, in this case, the system.

Thirdly, if the system is in the process of pursuing a session-goal, as by asking a series of questions, it needs to have a representation of the information that it has not asked for yet. This list is the value of the session-goal slot in the discourse position. In the normal course of pursuing such a session-goal the next item on the list will become the system's next current speech act.

Finally, the system needs to have the content of its next utterance ready before it can determine which role to take in order to communicate it. This content we represent as one or more speech acts, including the type of speech act and the proposition, all of which we label the current-speech-act. This content will be produced by one or more of the system's plans either of the utterance-level or session-level types.

To summarize, the *discourse position* of a conversant is comprised of the following elements and is maintained by the conversant (in this case, the system):

1. **role**: which may be one of either the initiator or the responder.
2. **topic**: by this we mean a local subtopic whose change may be marked by the linguistic means suggested by Reichman [11], for instance, and corresponding the concept of global focus of Grosz [5].
3. **current speech act**: this is the goal of the current utterance of the conversant and may be: to answer a question, to ask a question, or other speech act(s).
4. **current session goal**: if the conversant is the initiator he may also be pursuing some extended domain-level goal consisting of several steps, of which the current speech act forms the current step. The remaining steps yet to be performed are the current session goal.

Using the information in the discourse position we can articulate an algorithm for managing initiator/reactor shifts in problem-solving dialogue. The relationship of the discourse position to the algorithm in execution may be made clearer by noting that the sequence of processing and responding to input is this:

1. user's utterance comes into advising system
2. utterance-level plans and session-level plans applied to user's utterance (after parsing). Some plan will always be activated, such as the default *answer* utterance plan.
3. activated system plan processes input and produces appropriate response which becomes the value of the current-speech-act in the system's discourse position.
4. if the activated plan has more than one step, such as a sequence of questions to be asked, then the remainder of the steps (after the first, which becomes the current-speech-act) becomes the current value of the current-session-goal slot in the discourse position.
5. the discourse management algorithm of figure 1 is applied that decides what shift in initiator/reactor roles is indicated given the current role and the current-speech-act. The nature of the shift will dictate which linguistic markers, if any, should be combined with the current-speech-act and sent to the generation module to derive the appropriate surface text.

The discourse management algorithm will make a choice of actions for the system to take. The actions available depend on the system's current role. The options open to the initiator vis-a-vis his role are:

1. continue: for instance, to the next step of an extended session-goal
2. abandon: if the initiated goal is complete
3. yield: to an attempt by the reactor to seize the initiator role. "Right" is a common linguistic marker used to indicate acquiescence to the imposition of the reactor role on the former initiator.
4. suspend: yield initiator role to the reactor, but with the expectation of returning directly to it and the current goal

The options open to the reactor are:

1. continue: by back-channel responses, or answers to questions, for instance.
2. attempt to seize the initiator role

The algorithm for making these choices is found in figure 1.

The logic of the algorithm proceeds in this way. If the system is in the initiator role it first checks to see if the last answer or response from the user needs clarification. If not, then it checks to see if its current speech act is "answer" which may happen if the user has just asked a question. If that is not the case then the system will check to see if it is in the process of pursuing a session-goal. If there is a session-goal it will continue with the next step in it. If there is not it will try to get a new session goal if it can. If no new session goal is found it will relinquish the initiator role and be ready to accept the reactor role.

If the system finds itself in the reactor role and its processing of the last utterance of the user has made the current-speech-act to be "request-information" or "inform", then it will try to take the initiator role to accomplish that speech act. It will act similarly for a communication goal such as, "clarify" or "correct". Lacking such a reason for taking the initiative it will continue in the reactor role.

This algorithm is partially implemented in the current ADVISOR project. We have implemented in detail a single case of one of the occasions to shift from the reactor role to the initiative role. We present this case of taking the initiative in the next section.

```
If role = "initiator"
  then if
    current-speech-act = "clarify" or "correct"
                        ⇒ CONTINUE, but interrupt current
                        goal to ask question or inform

    current-speech-act = "answer"
                        ⇒ YIELD to reactor's attempt to
                        seize initiator role
                        mark role = "reactor"

    current-session-goal is nil
                        ⇒ ABANDON initiator role
                        mark role = "reactor"

    else                ⇒ CONTINUE with next step
                        update current-speech act from
                        session goal

else if role = "reactor"
  then if
    current-speech-act = "request-information" or "inform"
                        ⇒ SEIZE initiator role
                        mark role = "initiator"

    current-speech-act = "clarify" or "correct"
                        ⇒ SEIZE initiator role
                        mark role = "initiator"

    else                ⇒ CONTINUE reactor role
                        current-speech-act = "answer"
```

Figure 1: Discourse Management Algorithm

3 A Detailed Case: Taking Initiative for Conflict in Goals

We now consider in detail a case of the subproblem of taking the initiative to provide unrequested information due to the detection of a conflict in goals between ADVISOR and the user. One of the responsibilities of an advising system would be to inform the user when he appears to be ready to do something in violation of the relevant rules. In the student advising domain this behavior is frequently observed. The student may be unaware of some departmental requirement, for instance. When the faculty adviser recognizes the gap in the student's knowledge, often by indirect inference, it is incumbent upon him or her to provide the relevant information to the student. Consistent with Grice's Maxim of Completeness the student expects intervention by the faculty member and construes its absence to be certification of the appropriateness of his intended plans.

The RENDEZVOUS problem appears in this context since there is a danger that by exercising too vigilantly its responsibility to warn of possible rule violation the system runs the risk of dominating the conversation excessively and unproductively. For instance, suppose a student merely asked about the content of a course for which he had not satisfied the prerequisites, but without implying that he intended to take it immediately. In this case it would be inappropriate for the system to alert the student that he had not taken the prerequisite course. The system needs to decide when an intervention is called for and when the evidence is insufficient to justify it. ADVISOR's method for doing so is based on an approximation of its strength of belief in various aspects of the violation condition.

3.1 The ADVISOR System

The ADVISOR system is an experimental dialogue system that functions as a faculty advisor to an undergraduate computer science major. It is basically a question-answering system with an underlying database of knowledge about the courses offered in the computer science department and a mini-expert system that can reason about choosing courses. It is currently capable of processing English input and producing an English response in some cases, although in the case we will consider its output is not English text, but a deep structure representation of the output.

ADVISOR parses the input from the user with an ATN parser that uses Woods-type [15] templates for the semantics. The input is parsed both into a speech act representation and into the form of a transaction against the database of domain knowledge represented as a KL-ONE net. The speech act formulation serves as the basis of a goal inferencing method that is derived from Perrault and Allen [10] and Carberry [2]. This inferencing method allows the system to make pragmatic inferences as to the goals of the user as expressed indirectly in the dialogue. For instance, if the user asks, "Who is teaching artificial intelligence this semester?" the goal inferencing mechanism will be able to derive the plausible inference (among others) that the user may want to take the artificial intelligence course. The output of the goal inferencing module is the set of plausible goals the user may be pursuing each of which is represented as a plan with a body, preconditions, and effects. This output goes to the discourse manager that consists of a set of the system's own plans and code to implement the algorithm for managing the initiator/reactor roles. The discourse manager can also use the KL-ONE retrieval function produced by the parser to make transactions against the KL-ONE knowledge base. Questions of the type, "Should I take operating systems this semester?" or "Can I take numerical analysis?" are handled separately from the others. For these questions a mini-expert system, or rule-based inference engine is called to reason about choice-making in the domain. If called, this module sends its output to a generator, written in PROLOG, that produces the surface English. If the question is not of that type then the discourse manager module produces the deep structure of the response. Eventually, the output of the discourse manager will go to the generator also. In the system diagram all of the links are functional at present except for the line from the discourse manager to the surface generator.

3.2 Responding to a Conflict in Goals: An Example

We now consider an example from the operation of ADVISOR in which the system detects a conflict between the user's goal and its own by means of an utterance-level system plan. The system, initially in the role of reactor responding to the user's questions, must decide among its options of continuing to answer the questions, answering while pointing out the conflict, or taking the initiative and addressing the conflict directly. On the basis of an evaluation of the strength of its belief in the violation it will choose the appropriate one among a set of responses to becomes its new current speech act. Here we have assumed a segment of discourse about a single topic and so have not implemented a method of monitoring and shifting topic. In this example ADVISOR does not have a current-session-goal.

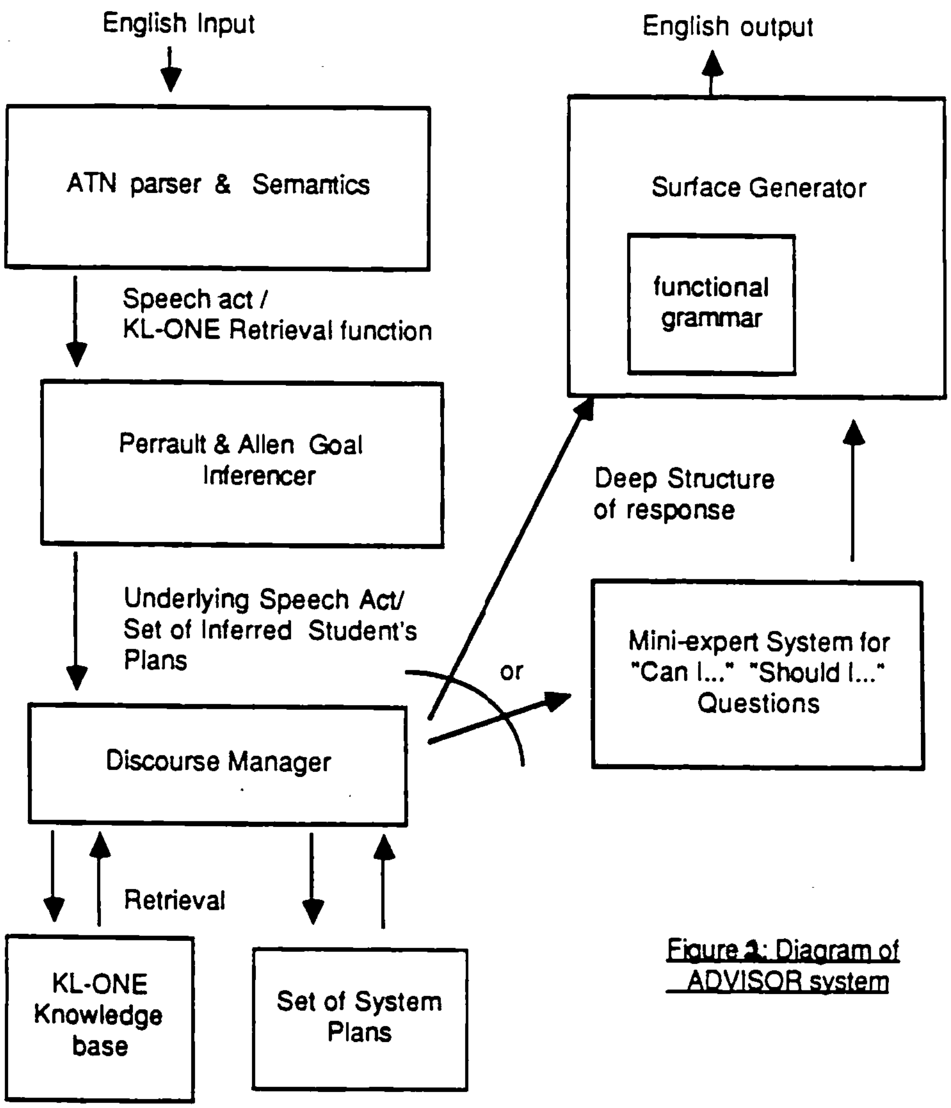

Figure 2: Diagram of ADVISOR system

In detecting an apparent conflict between the user's goals and the system's, it is important that the system have a set of choices ranging from taking the initiative to clear up the apparent problem to doing nothing at all, possibly because the violation may not be important or very likely. If a system were to intervene on every possible occasion it would quickly frustrate the user into avoiding the system. The relevant system resources utilized in this mini-dialogue include an utterance-level plan called *check-prerequisites* which attempts to verify that a student has satisfied the prerequisites for a course that he appears to be interested in taking. In order to be able to make inferences about such things as when a student intends to take a course without saying so explicitly, the system has a database of likely student plans to which it applies an inferencing method after Perrault and Allen [10]. This inferencing method has been extended to produce inferences which are sometimes definite as well as the plausible ones produced by Perrault and Allen's rules. (For a fuller description of this aspect of the system see [8].) For checking prerequisites it has a list of courses the student is known to have taken although it does not assume the list to be complete. Depending on the contents of the list ADVISOR can decide that the belief that a student has not taken a particular prerequisite course, i.e. the "violation," is definite or plausible.

Since ADVISOR's belief that the student is actually pursuing the inferred plan and the belief that he has violated a precondition of the plan can both be either "definite" or "plausible" the various possibilities can be ordered in this way from weakest to strongest:

Case I: plan = plausible
violation = plausible
< Case II: plan = definite
violation= plausible

< Case III: plan = plausible
violation = definite
< Case IV: plan = definite
violation = definite

Figure 3: Ordering of Intervention Cases
According to Strength of Belief

The available responses can also be ordered by strength as shown in figure 4.

Answer only < Answer + Question < Question only

< Answer + Warning < Warning only

Figure 4: Ordering of Possible Responses

Our solution is then to map the responses to the cases in the manner shown in figure 5. The relative strength of the response is correlated to the strength of the system's belief that a violation has occurred.

Case I: Answer only
Case II: Answer + question *or* Question only
Case III: Answer + warning
Case IV: Warning only

Figure 5: Responses Tailored to Situation

Examples for each of the cases have been implemented. Figure 6 shows ADVISOR's output for an example of CASE II. Initially ADVISOR's discourse position shows that it is in the role of reactor and its (default) current speech act is to answer. It has no current-session-goal and the topic is (and will continue to be) prerequisites. The student's question about who is teaching nlp would be sufficient for the system to infer that the student plausibly has the goal of selecting nlp. Since he announces this goal explicitly it becomes definite. When the student's plan of selecting a course is inferred ADVISOR's utterance-level plan *check-prerequisites* fires. The check-prerequisites plan tries to determine whether, in this case, the student has taken ai, the prerequisite to nlp. It can not determine whether he has or not and so marks the violation, the failure to take ai, as plausible. It can now determine that the situation corresponds to CASE II and chooses the "answer + question" option for ADVISOR's next current speech act. Since its new current speech act involves asking a question the system must now take the initiative and so it marks the role slot in its new discourse position as initiator and updates the current speech act slot. The other slots do not change. The current speech act would then be passed to the surface generator for transformation into English. (Since the interface between the discourse manager and surface generator has not yet been established ADVISOR's only output in this case is a deep structure representation of the response for which we have supplied a gloss in Figure 6.)

By contrast if the student had asked simply, "Who is teaching nlp this semester?" The same inferences about plan and violation would have occurred, but with a belief-level in each case of plausible. A response including a question about prerequisites would therefore be inappropriate. ADVISOR would

ADVISOR's prior knowledge: list of courses taken by student does not include artificial intelligence which is the prerequisite for nlp.

ADVISOR's current discourse position:
```
(role reactor)
(topic prereqs)
(current-speech-act answer)
(current-session-goal nil)
```

student: I want to take nlp. Who is teaching nlp this semester?

inferred goal: (select c:nlp)
strength of belief in inferred goal: definite
strength of belief in violation of prerequisite: plausible

ADVISOR's new discourse position:
```
(role initiator)
(topic prereqs)
(current-speech-act ((answer) (conj but)
        (askif (taken (agent user)
               (object c:ai)))))
(current-session-goal nil)
```

ADVISOR: ((answer) (conj but) (askif (taken (agent user)
(object c:ai))))

;; Mckeown, but have you taken ai?

Figure 6: ADVISOR system output
Showing Taking of Initiative: Case II

in that case only answer the question.

4 Future Work

Having shown an example of initiative-taking for an utterance plan, we would like to add one or more session-level plans. Such a plan will determine when and how the system should take over the dialogue in a more extended way. One idea is to have a default schema for the overall structure of the session and base interventions on deviations from the default schema.

In addition to taking the initiative the model of role handling should be developed toward complete manipulation of the initiator and reactor roles including seizure, maintenance, surrender, and abandonment of each role by implementing fully the algorithm of figure 1. Such a completely realized capability must include some way to stack dialogue states for later return.

We also plan to continue to analyze the data for a complete inventory of linguistic markers and their usage in role management.

All of these extensions are to be demonstrated in a fully implemented system.

5 Conclusions

We have shown why it would be desirable for certain computer systems to have the ability to interact in a more flexible fashion with the user than is now possible. One of the most important conversational abilities now lacking even in experimental systems is the ability to shift from the reactor role to the initiator role and back again in order for the system to accomplish its own goals related to helping the user. We have characterized these roles based upon observations of transcripts of faculty/student advising sessions. We then described what information would be necessary for an automated system to manage shifts in the roles of initiator and reactor. Finally, we presented in detail the case in which our experimental system ADVISOR takes the initiative because it detects a conflict between its goals and those of the student-user.

6 Acknowledgments

The author wishes to express his gratitude to Prof. Kathleen McKeown, Columbia University, New York, New York, for the benefit of her continuing guidance throughout this and related work, and to Dr. Myron Wish, A.T. & T. Bell Laboratories, Murry Hill, NJ, for helpful discussions on an earlier version of this paper.

References

1. Allen, J.F. "Argot: A System Overview". *Comp. & Maths. with Appl. 9*, 1 (1983), 97-109.

2. Carberry, S. Tracking User Goals in An Information Seeking Environment. Proceedings of the National Conference on Artificial Intelligence, 1983.

3. Carbonell, J.G.. *Subjective Understanding: Computer Models of Belief Systems.* UMI Research Press, Ann Arbor, Michigan, 1981.

4. Codd, E. F., et al. Rendezvous Version 1: An Experimental English-Language Query Formulation System for Casual Users of Relational Databases. IBM Research Laboratory, San Jose, Ca., 1978. Technical Report# RJ2144(29407).

5. Grosz, B. J. The Representation and Use of Focus in Dialogue Understanding. 151, SRI International, Menlo Park, Ca., 1977.

6. Kaplan, S. J. *Cooperative Responses from a Portable Natural Language Database Query System.* Ph.D. Th., U. of Penn., Phila.,Pa., 1979.

7. Lehnert, W. G.. *The Process of Question Answering.* Lawrence Erlbaum Associates, Hillsdale, N. J., 1978.

8. McKeown, K. R., Wish, M. and Matthews, K. Tailoring Explanations for the User. Proceedings of the IJCAI, 1985.

9. Pazzani, M. J. Interactive Script Instantiation. Proceedings of the National Conference on Artificial Intelligence, 1983.

10. Perrault, C. R., & Allen, J.F. "A Plan-Based Analysis of Indirect Speech Acts". *Am. Journal of Comp. Ling. 6*, 3-4 (July-Dec. 1980).

11. Reichman, R. Conversational Coherency. Harvard Univ. & Bolt Beranek and Newman Inc., Cambridge, Ma., 1978. Technical Report# 95.

12. Reichman, R. Plain Speaking: A Theory and Grammar of Spontaneous Discourse. Bolt Beranek and Newman Inc., June, 1981. Technical Report# 4681.

13. Schank, R.C., & Lehnert, W. The Conceptual Content of Conversation. Yale University, New Haven, Conn., 1979. Technical Report# 160.

14. Webber, B.L., & Finin, T.. *In Response: Next Steps in Natural Language Interaction.* Ablex Publ. Co., Norwood, NJ, 1984.

15. Woods, W.A. Progress in Natural Language Understanding-- An Application to Lunar Geology. Proceedings of the National Computer Conference, 1973.