Analyzing Intention in Utterances*

James F. Allen

Computer Science Department, The University of Rochester, Rochester, NY 14627, U.S.A.

C. Raymond Perrault

Department of Computer Science, The University of Toronto, Toronto, Canada

Recommended by N.S. Sridharan

ABSTRACT

This paper describes a model of cooperative behavior and describes how such a model can be applied in a natural language understanding system. We assume that agents attempt to recognize the plans of other agents and, then, use this plan when deciding what response to make. In particular, we show that, given a setting in which purposeful dialogues occur, this model can account for responses that provide more information that explicitly requested and for appropriate responses to both short sentence fragments and indirect speech acts.

1. Introduction

A good question answering system often needs to provide a response that specifies more information than strictly required by the question. It should not, however, provide too much information or provide information that is of no use to the person who made the query. For example, consider the following exchange at an information booth in a train station.[1]

(1.1) *patron*: When does the Montreal train leave?

(1.2) *clerk*: 3:15 at gate 7.

* This research was supported in part by the National Research Council of Canada under Operating Grant A9285. Preparation of the document was supported in part by the Alfred P. Sloan Foundation under Grant No. 78-4-15.

[1] All the examples given in this paper are taken from transcripts of dialogues collected at the information booth in Union Station, Toronto [12].

Artificial Intelligence **15** (1980), 143–178

Although the departure location was not explicitly requested, the clerk provided it in his answer.

Other examples of helpful behavior, however, do not involve language. For example, if the patron approached a closed door in the station carrying a large bag of groceries in each arm, the clerk might very well open the door for him. This may occur without any communication occurring between them. We claim that the motivation for the clerk's behavior in both these examples is the same; the clerk wants to assist the patron in furthering his goals.

This paper concerns the modeling of such helpful behavior and, in particular, it investigates how such a model can be used to explain several aspects of linguistic behavior. We make the following assumptions:

— People are rational agents who are capable of forming and executing plans to achieve their goals.
— They are often capable of inferring the plans of other agents from observing that agent perform some action.
— They are capable of detecting *obstacles* in another agent's plans.

Obstacles are goals in the plan that the other agent cannot achieve (easily) without assistance.

One form of helpful behavior arises because the observing agent assumes a goal of overcoming an obstacle in the other agent's plan. The plan to achieve this goal will involve the helpful actions.

Language can be viewed as an instance of such goal-oriented behavior. Utterances are produced by actions (*speech acts*) that are executed in order to have some effect on the hearer. This effect typically involves modifying the hearer's beliefs or goals. A speech act, like any other action, may be observed by the hearer and may allow the hearer to infer what the speaker's plan is. Often a speech act may explicitly convey a goal to the hearer that is an obstacle. For example, utterance (1.1) conveys to the hearer that the speaker needs to know the departure time of the train. But there may be other goals in the plan that were not explicitly conveyed but are also obstacles. The helpful response will attempt to overcome these obstacles as well as the explicitly mentioned ones.

This model provides the mechanisms to explain some interesting aspects of language, including:

— the generation of responses that provide more information than required (as in the above example);
— the generation of responses to sentence fragments;
— the analysis of indirect speech acts.

Let us consider each of these in turn.

It is fairly simple to see how the model explains the providing of more information than explicitly requested. In the train domain, the clerk expects that the patron has goals such as boarding and meeting trains. The query about a train departure time (i.e., (1.1)) indicates that it is likely that the patron's plan is to board the train. In addition, assuming that the clerk believes that the patron does not already know the departure location, he believes knowing the location is also an obstacle in the plan. Thus he generates a response that overcomes both obstacles (i.e., (1.2)).

Another problem that arises in the transcripts is that people often communicate using sentence fragments. For instance:

(2.1) *patron*: The 3:15 train to Windsor?

(2.2) *clerk*: Gate 10.

Neither the syntactic form of the query nor the meaning of its words indicate what the response should be. However, in the model above, it is quite conceivable that the information in the fragment is sufficient to allow the hearer to deduce what the speaker's plan is. Hence he can produce a reasonable response based on what the obstacles in the plan are. In the above example, (2.1) is sufficient to identify the speaker's goal to board the 3:15 train to Windsor. An obstacle in this plan is knowing the departure location, hence the response (2.2).

Other sentences in the dialogues are not treated at face value. For instance,

(3.1) *parton*: Do you know when the Windsor train leaves?

Syntactically, this is a yes/no question about the hearer's knowledge. However, an answer of 'yes' in the given setting would be quite inappropriate. In other surroundings, however, it could be intended literally. For instance, a parent seeing a child off at the station and wanting to make sure that everything is arranged might say (3.1) intending to receive a yes/no answer. Sentences such as this that appear to mean one thing yet are treated as though they mean something else are termed *indirect speech acts* [24]. With relatively minor extensions to the basic plan inference/obstacle detection model, these forms can be explained. Intuitively, the solution lies in the realization that the speaker knows that the hearer performs such helpful behavior, and hence may say something intending that the hearer infer the indirect goal. There is not space to examine this problem in depth here; it is considered in detail in [16].

This paper describes the plan inference and obstacle detection processes and shows how they can be applied to explain helpful responses and the understanding of sentence fragments. Section 2 provides an overview of the general methods and Section 3 gives more details of the processes.

The remaining sections apply these techniques to analyzing language and provide some examples. Section 4 considers responses that provide more information than asked for and Section 5 considers the analysis of sentence fragments. Section 6 considers the generation of subdialogues intended to clarify the intent of a previous utterance.

The system described in this paper has been implemented and tested in the train domain described above [1]. While the dialogues in this domain are somewhat restricted in subject matter, they provide a wide range of interesting linguistic behavior.

2. An Overview of the Model

Let us start with an intuitive description of what we think occurs when one agent A asks a question of another agent B which B then answers. A has a *goal* to acquire some information; he creates a plan (*plan construction*) that involves asking B a question whose answer will provide the information. A then executes his plan, asking B the question. B receives the question, and attempts to infer A's plan (*plan inference*). In this plan, there may be goals that A cannot achieve without assistance. These are the *obstacles* in A's plan. B can accept some of these obstacles as his own goals and create a plan to achieve them. B's response is generated when he executes this plan.

This section outlines the mechanisms that are needed to specify this model more precisely. The first part of it considers the issue of representing knowledge about the world, goals, actions and speech acts. The succeeding sections describe the plan construction, plan inference, and obstacle detection processes, respectively.

2.1. Actions, plans and speech acts

We need to be able to represent our intuitive notions of plan, goal, and action and relate them to language. These problems have already received much attention, both as problems in the philosophy of language and from the point of view of artificial intelligence.

Existing problem solving systems provide a formulation of actions and plans [7, 8]. In these systems, the world is modelled as a set of *propositions* that represent what is known about its static characteristics. This world is changed by *actions*, which can be viewed as parameterized procedures. Actions are described by *preconditions*, conditions that must hold before the action can execute, and by *effects*, the changes that the action will make to the world. Given an *initial world state* W and a *goal state* G, a *plan* is a sequence of actions that transforms W into G.

Austin [2] suggested that every utterance is the result of several actions or *speech acts*. We are particularly interested in speech acts such as requesting, warning, asserting, and promising. These speech acts are appropriate only in certain circumstances. In particular, they may require the speaker and the hearer to have certain beliefs and intentions (wants). For example, to sincerely *INFORM* you that I am tired, I must believe that I am tired and I must intend to get you to believe that I am tired. Both these conditions can be modelled as preconditions on the INFORM act. A simple version of this act could have the effect that you now believe that I am tired.

Cohen [5] demonstrated that speech acts such as request and inform can be modelled successfully as actions in a planning system. He showed how speech acts may be planned in order to achieve specific (typically non-linguistic) goals.

2.2. Plan construction

Given a goal state, two major tasks need to be done to produce a plan to achieve that goal. One is to find a sequence of actions that will accomplish the transformation from the initial world state to the goal state. The other concerns specifying the bindings for the parameters of the actions in the constructed plan.

A typical method of constructing a plan is backwards chaining: given a goal G, find an action A that has G as one of its effects. Then evaluate the preconditions of A. If some of these conditions are not satisfied in the initial state, they become subgoals and the plan construction process repeats.

Another dimension of plan construction involves planning at different *levels of abstraction*. For example, in a domain where a robot has to plan a route through many rooms, the plan would first be developed in terms of 'go to room x' and 'open door y'. Only after such a plan was constructed would one consider planning actions such as 'rotate n degrees', propel forwards', 'twist arm', etc. To incorporate this, many actions must have the capability of being 'broken down' into sequences of more specific actions.

In order to facilitate reasoning about the planning process, we characterize it as a set of planning rules and a control strategy. Since this paper is mainly concerned with plan inference, we will not consider control strategies for planning explicitly. However, many of the control issues for plan inference are directly applicable to planning as well.

The planning rules are of the form 'If agent A wants to achieve X, then he may want to achieve Y', and are written:

$$\mathrm{AW}(X) = c \Rightarrow \mathrm{AW}(Y).$$

For example, a simple rule is:

(C.1) If an agent wants to achieve a goal E, and ACT is an action that has E as an effect, then the agent may want to execute ACT (i.e., achieve the execution of ACT).

One other rule of interest concerns reasoning about knowledge necessary to execute an action.

(C.2) If an agent wants to achieve P and does not know whether P is true, then that agent may want to achieve 'agent knows whether P is true'.

These ideas will be made more precise in Section 3.

2.3. Plan inference

Plan inferencing concerns the attempted (re)construction of some other agent's plan based on actions that that agent was observed performing. This process depends on both the observer's knowledge of how plans are constructed and on his original beliefs about what goals the other agent is likely to have.

The plan could be inferred in two manners. Starting from the expected goals, the observer could simulate the other agent's planning process, searching for a plan that includes the observed action. Obviously, most of the time such an approach is impractical. The alternative is to reconstruct the plan from the observed action, effectively applying the plan construction rules in reverse. The method we propose depends mainly on the latter approach, but does allow the possibility of the former occurring when circumstances permit. Note that, in actual fact, people probably use much more specialized knowledge to infer the plans of other agents, thereby bypassing many of the particular inferences we will suggest. Our approach so far, however, has been to specify a minimal set of reasoning tools that can account for the behavior observed. Given these tools, we then hope to precisely define and explain the more complex and specialized mechanisms by deriving them from the simple set.

As with the plan construction process, the plan inference process is specified as a set of inference rules and a control strategy. Rules are all of the form 'If agent S believes agent A has a goal *X*, then agent S may infer that agent A has a goal Y.' Examples of such rules that correspond to the planning rules (C.1) and (C.2) are:

(D.1) If S believes A has a goal of executing action ACT, and ACT has an effect *E*, then S may believe that A has a goal of achieving *E*.

(D.2) If S believes A has a goal of knowing whether a proposition *P* is true, then S may believe that A has a goal of achieving *P*.

Of course, given the conditions in (D.2), S might alternately infer that A has a goal of achieving *not P*. This is treated as a separate rule. Which of these rules applies in a given setting is determined by control heuristics.

The plan inferencing involves a search through a set of partial plans that consist of two parts: one is constructed using the plan inference rules from the observed action (and called *alternatives*), and the other is constructed using the plan construction rules on an expected goal (and called *expectation*). When mutually exclusive rules can be applied to one of these partial plans, the plan is copied and one rule is applied in each copy. Each partial plan has *tasks* associated with it that attempt to modify and further specify it. Typical tasks involve application of the rules and identification of the referents of variables in the plan.

Partial plans are rated as to how probable they are to be the correct plan. This rating is determined using a set of heuristics that fall into two classes:

those that evaluate how well-formed the plan is in the given context and those that evaluate how well the plan fits the expectations. An example of a heuristic is:

(H1) Decrease the rating of a partial plan if it contains a goal that is already true in the present context.

The tasks associated with a partial plan are rated according to the plan's rating and compete for execution according to this rating.

2.4. Obstacle detection

We claim that many helpful responses arise because the hearer detects obstacles in the speaker's plan, i.e., goals that the speaker cannot (easily) achieve without assistance. The most obvious obstacles are those that the speaker specifically brings attention to by his utterance. These *explicit obstacles* are goals that are an essential part of the chain of inferences that the hearer makes when he deduces the speaker's plan. For example, to infer the plan of the speaker from the utterance

(4.1) When does the Windsor train leave?

the hearer must infer that the speaker has the goal of knowing when the train leaves. Thus, this is an explicit obstacle.

However, the hearer cannot base his response solely on these explicit obstacles. For instance, if A, carrying an empty gas can, comes up to S on the street and asks

(5.1) Where is the nearest gas station?

and S answers

(5.2) On the next corner.

knowing full well that the station is closed, then S has not been helpful. But S's response did overcome the explicitly mentioned obstacle, namely knowing where the nearest gas station is. S may want to notify A if there are other obstacles to his plan, especially ones that A is not aware of. This behavior is expected; even if A and S are strangers, if A believes that S knew all along that the gas station was closed, he has justification for being angry at S, for S has violated some basic assumptions about human cooperation.

In the dialogues we have studied, all obstacles are caused by a lack of some information required in order to be able to execute the plan. This is not the case in general, as we saw in the example where the clerk opens the door for the patron carrying the groceries. There the obstacle concerned the patron's inability to physically perform the action of opening the door.

2.5. Related work

Although there has been some previous work on recognizing plans and generating helpful responses, to our knowledge, no one else has attempted to combine the two techniques. Bruce [4] outlines a general model of story comprehension based on recognizing the intentions of the characters in the story as well as the intentions of the author. Although a slightly different application, our work here agrees quite closely with his view, and it will be interesting to attempt to combine the approaches. Bruce does not, however, describe any algorithm for actually recognizing the intentions in his stories.

Schmidt et al. [23] discuss a plan recognition algorithm where physical actions are observed and their task is to discover what the agent is doing. But they allow an arbitrary number of acts to be observed before committing themselves to a particular plan. This technique is appropriate for analyzing sequences of actions. In our work, however, it is essential that we identify the speaker's plan from a single observed action (i.e., his utterance).

The system PAM [25] analyzes stories by constructing a plan for the participants and then answers questions about the story using the plan. However, it does not attempt to recognize the plan of the agent asking the questions i.e. it does no plan based reasoning about how language is used. PAM answers questions solely on the form of the question asked (see [14]).

Kaplan [13] discusses some helpful responses to questions. But these are based on violated presuppositions conveyed by the question. He does not discuss any mechanisms that could explain helpful behavior in its general form.

The most important distinguishing characteristic of our research is that we emphasize the use of a model of the other's beliefs and goals. Helpful responses are detected because of this ability to reason about the other agents. A few more specific comparisons on linguistic issues will be discussed as the work is presented.

3. Plan Inference and Obstacle Detection

Some representation issues must be considered before the plan inference process can be described. Section 3.1 discusses the representation of belief, knowledge and want, and Section 3.2 considers actions and plans. The description of plan inference is then broken into three parts: we consider the plan inference rules in Section 3.3, the rating heuristics in Section 3.4, and the control of the process in Section 3.5. The final section considers how obstacles are detected in the plans that are inferred.

3.1. Belief, knowledge, and wants

An adequate description of modeling belief would take an entire paper by itself. We can just outline a few of the important issues here. Our treatment of belief is virtually identical to that of Hintikka [11]. The reader interested in the representation should see Cohen [5].

The crucial property of belief is that what one agent S believes another agent A believes has no logical relation to what S believes. Thus, S may believe A believes the world is flat while personally believing that it is round.

Intuitively, the belief operator allows us to consider actions and plans from another agent's point of view. This can be captured crudely by the axiom schema

$$(\mathrm{AB}(P \Rightarrow Q) \ \& \ \mathrm{AB}(P)) \Rightarrow \mathrm{AB}(Q).$$

Thus, S may infer that A inferred some proposition Q if S believes that A believes that there is sufficient evidence to infer Q. We also need an axiom that states that conjunction can 'pass through' the belief operator. Thus, if S believes that A believes P is true and that A believes Q is true, then S also believes that A believes P and Q are true, and vice versa. Written more formally:

$$\mathrm{AB}(P) \ \& \ \mathrm{AB}(Q) \Leftrightarrow \mathrm{AB}(P \ \& \ Q).$$

A similar axiom is *not* valid for disjunction.

Note that to be completely adequate, the belief operator B would have to be indexed by the time when the belief was held. For the sake of simplicity, however, we will ignore time throughout this paper.

Some formulas involving beliefs occur commonly enough to warrant special mention. In particular, there are three constructs associated with the word 'know' that arise very frequently.

The first involves representing that an agent S believes some A knows that P is true. This not only conveys the fact that S believes A believes P, but also that S believes P as well, i.e.,

$$\mathrm{SB}(P \ \& \ \mathrm{AB}(P)).$$

As an abbreviation, we define

$$\mathrm{A\ KNOW}\ P = P \ \& \ \mathrm{AB}(P).$$

In other words, if SB(A KNOW P), then S believes that S and A agree that P is true. This, of course, has no implication as to whether P is 'actually' true.

The next structure involves uses of 'know' as in 'John knows whether P is true'. This is the type of belief S would have to have if S believed that John was able to answer a question such as 'Is P true?'. It is represented as the disjunction

$$\mathrm{A\ KNOWIF}\ P = (P \ \& \ \mathrm{AB}(P)) \vee (\mathrm{not}P \ \& \ \mathrm{AB}(\mathrm{not}P)).$$

The final use of know is in the sense demonstrated by the sentence 'John knows where the box is'. This case is represented by quantifying over the B operator:

$$\text{A KNOWREF } D = (\exists y)\ (\text{the } x : D(x) = y)\ \&\ \text{AB}(\text{the } x : D(x) = y)$$

where 'the $x : D(x)$' is any description. In the above example it would be 'the x such that the location of the box is x'. For further details on these representations of 'know', see Allen [1].

Goals and plans of agents are indicated by using an operator want ('W'), i.e.,

$$\text{AW}(P) = \text{A has a goal to achieve } P.$$

By this, we mean that the agent A actually intends to achieve P, not simply that A would find P a desirable state of affairs. The properties of W in this work are specified completely by the planning and plan inference rules.

3.2. Actions and plans

As with the operators in STRIPS [8], actions can be grouped into families represented by *actions schemas*. An action schema consists of a name, a set of parameters and (possibly null) sets of formulas in the following classes:

Preconditions: Conditions that should be true if the action's execution is to succeed.

Effects: Conditions that should become true after the execution of the action.

Body: A specification of the action at a more detailed level. This may specify a sequence of actions to be performed, or may be a set of new goals that must be achieved.

Each action definition may also specify *applicability conditions* on the parameters: conditions that must be true for the action to be well-defined. Every action has at least one parameter, namely the *agent* or instigator of the action.

An *action instance* is a predicate constructed from an action schema name with a set of parameter instantiations and a time specification. The predicate is true only if the described action is (was or will be) executing at the specified time. For example,

> ACT(S)(t)—a predicate that is true only if the action ACT with agent S was/will be executed at time t.

We will say that an action is *intentional* if whenever the action was performed, the agent wanted it to occur at that time. Thus, if ACT is an intentional act, A any agent, t any time, then

$$\text{ACT(A)}(t) \Rightarrow \text{AW(ACT(A)}(t)).$$

Thus, in a loose sense, there is a 'precondition' on every intentional action that the agent must want to perform the action. We will sometimes refer to this condition as the *want precondition* of the act.

In general, the time specification will be omitted. If an action is within the immediate scope of the B operator, it is assumed to have a time specification in the past. If it is within the immediate scope of a W operator, it is assumed to have a time specification in the future.

Actions are not only reasoned about, sometimes they are executed. The execution of an action is specified either as *primitive* or by its body. If the body is a sequence of other actions, this sequence may be recursively executed. If the body is a set of new goals, plan construction must be initiated on the goals and then the resultant plan executed.

It will often be convenient to refer to an action and its associated preconditions, effects and body as a single unit. Such *action clusters* are action schemas with instantiated parameters.

A *speech act* is an intentional action that has as parameters a speaker (i.e., the agent), a hearer, and a propositional content, and whose execution leads to the production of an utterance. Their preconditions and effects are defined in terms of the beliefs and wants of the speaker and hearer. For the present, we will assume that the speech act intended by the speaker can be readily identified from the syntactic form of the utterance. This assumption, which is obviously incorrect, will be removed in the later sections of the paper concerning indirect speech acts and sentence fragments.

In its final form, a *plan* is a linear sequence of action instances that will map an initial world state into a goal state. But as Sacerdoti [20] points out, plans cannot easily be constructed in linear form. He uses a representation that imposes only a partial ordering on the actions, where the orderings are imposed only when necessary. We use a similar representation.

A plan can be represented as a directed graph with predicates (goals and actions) as nodes and labelled arcs indicating their interrelationships. These arcs implicitly specify a partial ordering on the actions. The *enable* arc links a proposition that is a precondition of an action to that action. Likewise, an *effect* arc links an action to a proposition that is its effect. The *know* arc links a KNOWIF or KNOWREF proposition to a proposition in a plan whose truth values cannot be determined unless the 'know' proposition is true. For example, the planner cannot achieve the goal

'planner at the location of *n*'

unless

'planner KNOWREF the location of *n*'.

To permit plans to be represented at varying levels of detail, plan structures themselves can be nodes in a plan. These 'plan' nodes represent the bodies of actions. The *body* arc links an action to a plan node that contains its body.

3.3. The plan inference rules

The plan inference process starts with an incomplete plan, usually containing a single observed action or an expected goal and attempts to fill in the plan. The possible additions that can be made are described in this section as a set of plausible inference rules. They are presented without any consideration of whether the inference is reasonable in a given setting, for whether or not a rule is applied depends on the likelihood that the new plan specification it produces is the actual plan. This is evaluated using the heuristics described in the next section.

The notation

$$\mathrm{SBAW}(X) = i\Rightarrow \mathrm{SBAW}(Y)$$

indicates that if S believes A's plan contains X, then S may infer that A's plan also contains Y. The possible rules can be divided into three broad categories: those that concern actions, those that concern knowledge, and those that concern planning by others.

3.3.1. *The rules concerning actions*

These rules arise from the model of how plans are constructed. Throughout this section, S refers to the agent that is inferring the plan of another agent A.

$\mathrm{SBAW}(P) = i\Rightarrow \mathrm{SBAW}(\mathrm{ACT})$ if P is a precondition of action ACT.

[Precondition-Action Rule]

Thus, if A wants to achieve some goal P, then A may want to execute an action ACT enabled by P.

$\mathrm{SBAW}(\mathrm{B}) = i\Rightarrow \mathrm{SBAW}(\mathrm{ACT})$ if B is part of the body of ACT.

[Body–Action Rule]

Thus, if A wants to execute an action B that is part of the execution of another action ACT, A may want to execute ACT.

$\mathrm{SBAW}(\mathrm{ACT}) = i\Rightarrow \mathrm{SBAW}(E)$ if E is an effect of ACT.

[Action–Effect Rule]

Simply, this says that if A wants to execute an action, then A wants the effects of that action.

$\mathrm{SBAW}(n\mathrm{W}(\mathrm{ACT})) = i\Rightarrow \mathrm{SBAW}(\mathrm{ACT})$ if n is the agent of the intentional action ACT. [Want–Action Rule]

This rule is based on the want precondition for intentional actions. Intuitively, this says that if A wants n to want to do some action ACT, then A may want n to do ACT.

3.3.2. *The rules concerning knowledge*

These inference rules indicate how goals of acquiring knowledge relate to goals and actions that use that knowledge. The first two rules reflect the fact that if A wants to know whether a proposition P is true, then it is possible that A wants to achieve a goal that requires P to be true (or requires P to be false). The third one indicates that A wants to know whether P is true in order to establish the identity of one of the terms in P.

$$\text{SBAW(A KNOWIF } P) = i \Rightarrow \text{SBAW}(P); \qquad \text{[Know-positive Rule]}$$

$$\text{SBAW(A KNOWIF } P) = i \Rightarrow \text{SBAW(not}P); \qquad \text{[Know-negative Rule]}$$

$$\text{SBAW(A KNOWIF } P(a)) = i \Rightarrow \text{SBAW(A KNOWREF the } x : P(x)). \qquad \text{[Know-value Rule]}$$

Of course, in any plan alternative, at most one of these rules can be correct. The decision as to which of these is correct, or that none of these is correct, is the responsibility of the heuristic evaluation of the plans produced by applying them.

The final inference rule about knowledge concerns goals of finding the referents of descriptions. It suggests that such a goal indicates that A has another goal that involves the referent.

$$\text{SBAW(A KNOWREF the } x : D(x)) = i \Rightarrow \text{SBAW}(P(\text{the } x : D(x))) \qquad \text{[Know-term Rule]}$$

where $P(\text{the } x : D(x))$ is a goal or action involving the description (or its referent).

Because of the vagueness in the resulting goal, this rule does not produce reasonable plans unless a specific goal or action of form $P(D)$ already exists in the expectations.

3.3.3. *The rules concerning planning by others*

The plan construction process can be described in the same manner as the plan inference process; as a set of rules that describe possible constructions, and a set of heuristics to evaluate the resulting plans. The plan construction rules are simply the inverses of the plan inference rules. Some examples are given below. X is the name of the agent doing the plan construction.

$$X\text{W(ACT)} = c \Rightarrow X\text{W}(P) \quad \text{if } P \text{ is a precondition of ACT.} \qquad \text{[Action–Precondition Rule]}$$

Thus if X wants to execute ACT, X must ensure that its preconditions are ture.

$$X\text{W(ACT)} = c \Rightarrow X\text{W}(B) \quad \text{if } B \text{ is part of the body of ACT.} \qquad \text{[Action–Body Rule]}$$

$$XW(E) = c \Rightarrow XW(\text{ACT}) \quad \text{if } E \text{ is an effect of ACT.}$$

[Effect–Action Rule]

$$XW(P) = c \Rightarrow XW(X \text{ KNOWIF } P).$$

[Know-Rule]

Thus, if X wants to achieve P but doesn't know whether P is true, X must find out whether P is ture.

When X constructs a plan involving the cooperation of another agent Y, X may depend on Y to do some plan construction as well. Thus, X might get Y to perform some action ACT by getting Y to have the goal of achieving ACT's effects. For example, assume that X wants to have a surprise birthday party for his roommate Y and needs to get Y out of the house. X says

'We need some beer.'

expecting Y to assume the goal of getting beer, and then construct a plan to get some. This involves leaving the house, the goal X had all along. Thus X has reasoned about Y's planning process. Crudely, this new planning inference rule can be described as

$$XW(YW(\text{'leave house'})) = c \Rightarrow XW(YW(\text{'get beer'}))$$

since

$$XB(YW(\text{'get beer'}) = c \Rightarrow YW(\text{'leave house'})).$$

Thus, if X wants Y to want to do ACT, he may achieve this by getting Y to want to achieve E, where Y's planning process will infer ACT as a way of achieving E. In general, we have the set of plan construction rules

$$XW(YW(P)) = c \Rightarrow XW(YW(Q)) \quad \text{if } XB(YW(Q) = c \Rightarrow YW(P)).$$

[Nested–Planning Rule]

These rules are of interest when it is assumed that there is no deceit between the agents, and both realize that the planning by the hearer was intended. Thus, a king might say

'It's cold in here.'

to a servant, expecting the servant to plan to make the room warmer.

But for the servant to understand the king's intention in the above example, he must recognize that the king's plan included planning by the servant. We can characterize inferences that construct these new plans as follows (reverting back to S as recognizer, A as the observed agent):

$$\text{SBAW}(\text{SW}(P)) = i \Rightarrow \text{SBAW}(\text{SW}(Q))$$
$$\text{if } \text{SBAB}(\text{SW}(P) = c \Rightarrow \text{SW}(Q)).$$

[The Recognizing Nested–Planning Rule]

3.4. Rating heuristics

As mentioned above, plan inferencing is accomplished by a search through a set of specifications of *partial plans* that consist of two parts. One part is constructed using the plan inference rules from the observed action (and called the *alternative*), and the other is constructed using the plan construction rules from an expected goal (and called the *expectation*). (In the implementation, partial plans may contain many expectations sharing one common alternative.)

Each partial plan is assigned a rating, which is determined using heuristics described in this section, that reflects how likely it is to be part of the 'correct' plan. These heuristics are based solely on domain-independent relations between actions, their bodies, preconditions and effects. The initial partial plans are given a rating of 1. The heuristics are expressed here only in terms of increasing and decreasing the ratings. The actual formulas are very simple and given in Table 1 at the end of the section. This is organized in this way to emphasize the fact that while rating changes in the indicated direction are essential to our model, we feel that some variation is possible in the actual figures.

Finally, before we give the heuristics, we must make the distinction between actions that are currently executing, awaiting execution (pending), and have executed. In particular, the observed action is considered to be currently executing, and any action which contains an executing action in its body is also considered to be executing.

(A) *Action based heuristics.* Generally, one expects agents to construct plans that they believe they are able to execute, and they execute them only to achieve goals that are not presently true. This gives us two rules:

(H1) Decrease the rating of a partial plan if it contains an action whose preconditions are false at the time the action starts executing.

(H2) Decrease the rating of a partial plan if it contains a pending or executing action whose effects are true at the time that the action commences.

Heuristic (H1) states that if the action is presently executing, then the preconditions must not have been false when the action was initiated. On the other hand, if the action is pending, then its preconditions must be achieved within the plan or must be achievable by a *simple plan*. As a first approximation to a simple plan, we define it as a hypothesized plan consisting of a single action whose preconditions are already true.

(B) *Expectation-based heuristics.* These heuristics favor those partial plans whose alternatives seem most likely to merge with its expectation.

(H3) Increase the rating of a partial plan if it contains descriptions of objects and relations in its alternative that are *unifiable* with objects and relations in its expectation.

The term unifiable is used here in the sense of the unification algorithm found in resolution theorem provers (see [15]). Thus, if an alternative involves a train description, those expectations that involve a (compatible) train will be favored. Similarly, if an expectation involves a relation such as arrival time, its alternative seems more favorable if it also involves an arrival time relation.

(C) *Search based heuristics.* The remaining heuristics involve evaluating which partial plans should be considered next from a search efficiency point of view. These measure how specific the plan fragment is becoming. A couple of heuristics relate to events that produce important specializations to the plan, i.e., identifying referents of descriptions and identifying the speech act.

(H4) Increase the rating of a partial plan if the referent of one of its descriptions is uniquely identified. Decrease the rating if it contains a description that does not appear to have a possible referent.

(H5) Increase the rating of a partial plan if an intersection is found between its alternative and expectation, i.e., they contain the same action or goal.

The final heuristic favors alternatives that have produced inferences that are well rated enough to be applied.

(H6) Increase the rating of a partial plan each time an inference rule is applied.

A particular partial plan will be extended until either it splits sufficiently many times to lower its rating, or the rating heuristics start to disfavor it.

Each partial plan has a *weight* that is used to calculate its rating. Heuristic (H3) adds a fixed factor of 5 for each similarity found. (The actual value 5 has no effect on the search except possibly for roundoff considerations in the rating calculation.) All other heuristics affect the weight by a multiplicative constant.

TABLE 1. The multiplicative factors for the heuristics

Heuristic	Description	Factor
(H1)	Preconditions false	0.5
(H2)	Effects true	0.5
(H4)	Referent identified	1.5
(H4)	Referent impossible	0.2
(H5)	Intersection found	1.5
(H6)	Inference rule applied	1.25

The total weight of all the partial plans is used in calculating each plan's rating. The rating of a plan P is simply the percentage of the total weight that plan P has. The actual values of the multiplicative factors are provided in Table 1.

3.5. The control of plan inferencing

As mentioned previously, partial plans are modified and refined by a set of programs (*tasks*) that are attached to them. When a task is *suggested*, it is given a rating that is strongly dependent on the partial plan that it is to manipulate and is placed on a priority list according to this rating. The top rated task is always selected for execution and removed from the list. This section describes the various types of tasks.

It is important for search efficiency considerations that there be extensive interactions between the expectations and alternatives. The information in an expectation may specify constraints on an alternative that restricts what possible inferences could be made from it, and vice versa. For instance, if both the expectation and alternative refer to a train, the trains are assumed to be identical (unless they contradict).

The initial set of partial plans consists of all pairings of an alternative containing only the observed action and one of the original expectations. To allow for the possibility of an utterance that does not fit an expectation, a partial plan is also constructed with a null expectation.

The actual tasks that perform the plan inferencing can be divided into three classes: those that specify the structure of the plans, those that identify objects on the plans, and those that control the search.

(A) *The plan specification tasks.* The plan specification tasks make additions to a plan hypothesis according to the inference rules discussed above. Alternatives are expanded using the plan inference rules and expectations are expanded using the planning rules. There are many occasions when mutually exclusive rules can be made. In such cases, copies of the partial plan are made and one rule is applied in each. When such a split occurs, the rating of the partial plan is divided between its successors.

This is performed by two tasks: *Infer* and *Expand*. *Infer* examines a local area of a plan and suggests possible inference rules that apply. *Expand* actually applies these rules to modify the partial plan. The processing is divided to allow explicit control of the 'fan-out' of the search: when an *Expand* is suggested from an *Infer*, its rating is determined by the rating of the partial plan it concerns, plus an estimate of the number of splits it will make. The greater the number of splits, the lower the rating (see Table 2). The relation between these two is set so that the copying (i.e., the splitting) will not be done until the newly

TABLE 2. The rating of tasks relative to the rating (R) of their partial plan

Task	Rating Formula
Infer	$0.75 * R$
Expand	$0.75 * R * f(n)$ where $f(n) = 1.25/n$, where n is the number of new partial plans to be created by the *Expand*
Identify	R
Accept	R

created partial plans would be sufficiently well-rated to produce tasks that are competitive on the agenda.

(B) *The Identify task*. The plan being inferred will usually contain many descriptions of objects whose referents must be identified. Some of these descriptions were introduced by the utterance while others are introduced by the inferences. For an example of the second type, say in applying the precondition–action rule, there is a parameter in the action definition that is not part of the precondition. Then this parameter will not be specified when the action is introduced into the plan, although there may be constraints on its referent imposed in the action definition.

Likewise, existing descriptions may acquire additional constraints as the inferences are made. An action introduced into a hypothesis may specify constraints on one of the parameters in the plan that unified with one of its parameters.

Thus new descriptions may be introduced and old descriptions may be further specified as the inferences are made. Each time such a change occurs, an *Identify* task is suggested that will attempt to find a referent.

Note that since some agent S is inferring another agent A's plan, all evaluation of descriptions must be done with respect to what S believes A believes. In general, if S believes that there is only one object that A believes could fit the constraints, then it is the referent. This is in fact not a sufficient condition, but will suit our purposes here. Perrault and Cohen [18] examine reference problems in detail.

Identification of referents may require the use of domain specific inferences. For example, in the train domain, there is a need for an inference rule that says, 'If a train is described without a time specification, then it is probably the next train that fits the description.' It remains a problem as to whether such a heuristic could be inferred from the general structure of plans, or whether it is truly domain specific.

(C) *The search control tasks.* Most of the control mechanisms are built into the rating scheme and the plan inferencing monitor. For instance, every time an addition is made to an alternative, the expectations are examined for new similarities caused by the addition. This may cause a change in the ratings according to the expectation-based rating heuristics.

Some mechanism must terminate plan inferencing. This is done by the task *Accept*, which is suggested by the monitor whenever an intersection of alternative and an expectation seems possible because they contain unifiable specifications of a step (i.e., an action or goal) in the plan, or when the plan with the null expectation is rated twice as favorably as the other partial plans. *Accept* must decide whether to terminate the plan inferencing or not. At present, the termination condition is fairly simple: if the plan under consideration is rated twice as favorably as any other partial plan, it is accepted. This is implemented by suggesting a dummy task at half the present task's rating. This task will sit on the pending list until all better rated tasks have executed. When it comes to the top, if no other *Accepts* have been executed, the original alternative is identified as the speaker's plan and plan inference stops. If another *Accept* has executed, there is an ambiguity (see Section 6).

3.6. Obstacle detection

Once S has inferred the speaker's plan, the next step, if he is to be helpful, is to identify the obstacles in that plan.

An obstacle is any goal specification that is not initially true or achieved within the plan. Some obstacles involving knowledge are only implicitly in the plan. For example, if a proposition P is a goal in A's plan, and S believes that A does not know whether P holds, i.e., SB(A *not* KNOWIF P), then A KNOWIF P is an implicit obstacle. Similarly, if the plan involves a description, say 'the $x:D(x)$', that S believes A does not know the referent of, then 'A KNOWREF the $x:D(x)$' is an implicit obstacle. These implicit obstacles can be derived by applying the knowledge plan inference rules to each step in the plan. Problems arise from the fact that the entire plan of the speaker may not be inferred, as only enough of the plan to link the observed utterance to an expectation is generated by the plan inference. So there may be apparent obstacles according to the partial plan that would not be an obstacle in the complete plan. One strategy to eliminate some of these is to not consider obstacles that can be easily achieved by a simple plan by the speaker.

There are other techniques for eliminating possible obstacles. The obstacles can be partially ordered using the ordering constraints imposed by the plan relations. In such cases, S knows he must address any obstacles prior to a given obstacle 0 if he is to address 0. For example, if A is carrying groceries in his arms and needs to pass through two doors, it does no good for S to open the second door unless he also opens the first.

Another effective filter on the obstacles involves considering which obstacles the hearer intended to communicate. In particular, the goals that (S believes) A believes S can achieve are most likely to have been intended by A. For example, in the train station setting, the clerk not only does not sell tickets, but he also believes that the patrons know this. As a consequence, although not having a ticket is an obstacle, in the plan of boarding a train, the clerk does not expect the patron to ask him for a ticket (because he can't provide one).

The above are useful strategies if S believes that both S and A agree on what the obstacles in the plan are. However, if S and A disagree on some issue, special obstacles occur that must be addressed. For example, if A thinks that state X already holds and is depending on X in his plan, but S believes X does not hold, then S is obliged to mention this fact to A. Otherwise, A's plan will fail and S will be considered as uncooperative. In the reverse case, if A thinks state X is not true, but S believes it in fact already holds, then S must tell A, for A will not execute his (valid) plan because he thinks it will not succeed.

There is one class of obstacle that is truly difficult to detect but should be considered. If there are two goals in a plan and one is just a step towards achieving the second, then the heuristics above will indicate that the first is the only obstacle. However, in some cases, achieving the second eliminates the need to ever (even temporarily) achieve the first. For example, if A and S are in a locked room and A asks S where the key to the door is, S might deduce the following goals:

'A know where key is' in order to 'Get the door open'.

If S opens the door himself, say by some means other than using the key, then the goal of knowing the key's location becomes irrelevant. However, detecting such situations is quite difficult and beyond the scope of the present work, for it may involve considering the speaker's plan to an arbitrary distance into the future with no well defined termination condition.

The algorithm used in the system involves testing every goal statement in the plan. Obstacles are selected using the following preferences:

(1) those goals that S and A disagree about whether they hold or not;

(2) those goals that are explicitly indicated as obstacles by the utterance, i.e., the inference path from the surface speech act to an expected goal includes the goal;

(3) those goals that are required to perform the actions that are partially enabled by the goals in class (2), but are not achieved in the plan; and

(4) those goals that are not 'preceded' by other goals in the partial ordering of obstacles, and are not achieved in the plan.

The algorithm produces a set of obstacles in the highest preference class that is not empty.

4. Examples of Helpful Responses

This section provides three examples of helpful responses that can be produced using the plan inference and obstacle detection processes. The first shows a simple response that provides more information than was explicitly requested. It is described in some detail to give an idea of the actual plan inference mechanisms in operation. The second examples considers a yes/no question and shows why extra information should be provided if the answer is negative. The third example shows how an appropriate answer may be given to a question that is an indirect speech act. This final example is only briefly sketched. A detailed account of indirect speech acts can be found in Perrault and Allen [16] or Allen [1].

Before the examples are presented, the train domain is specified (Section 4.1) and definitions are given for the speech acts (Section 4.2).

4.1. The train domain

The setting for the examples is the train station information booth, S is the system playing the role of the information clerk, and A is the patron at the station. The non-linguistic actions relevant to the simple train domain are:

BOARD(agent,train,station) : SOURCE(train,station)
precondition: AT(agent, the x : DEPART.LOC(train,x),
  the x : DEPART.TIME(train,x))
effect: ONBOARD(agent,train)
MEET(agent,train,station) : DEST(train,station)
precondition: AT(agent, the x : ARRIVE.LOC(train,x),
  the x : ARRIVE.TIME(train,x))
effect: MET(agent,train)

S's expectations are the partially instantiated plans formed from the following action instantiations with their preconditions and effects:

BOARD(A, ⟨train⟩, TORONTO)

MEET(A, ⟨train⟩, TORONTO)

where the angle brackets (⟨ · · · ⟩) indicate unspecified parameters in the expectations. At the start of processing, there are three partial plans each with a null alternative and each containing one of the BOARD, MEET, or NULL expectations. The first example shows that the number of expectations could be increased without greatly affecting the combinatorics of the search.

4.2. The speech act definitions

The speech act definitions provided here are very superficial. A more adequate account is found in Perrault and Allen [16]. The INFORM speech act, which is typically realized as a declarative sentence, is defined as follows:

> INFORM(speaker, hearer, *P*)
> want-precondition: speaker want INFORM (speaker, hearer, *P*)
> precondition: speaker KNOW *P*
> effect: hearer KNOW *P*

For an agent A to sincerely inform an agent H that *P* is true, A must believe that *P* is true (the precondition), and he must intend to get H to know that *P* is true (the effect of a successful inform). Note that this action cannot succeed without the cooperation of the hearer, for only he can change his own beliefs.

In many cases, agents reason about inform acts to be performed (by others or themselves) where the information for the propositional content is not known at the time of planning. For example, A may plan for S to inform A whether *P* is true; A cannot plan for S to perform INFORM(S,A,*P*) since this assumes that *P* is true. Thus we need two other 'views' of the INFORM act: INFORMIF and INFORMREF: (standard want-preconditions are omitted):

> INFORMIF(speaker, hearer, *P*)
> precondition: speaker KNOWIF *P*
> effect: hearer KNOWIF *P*

and

> INFORMREF(speaker, hearer, description)
> precondition: speaker KNOWREF description
> effect: hearer KNOWREF description

One further speech act that we need models one agent requesting another agent to do some action:

> REQUEST(speaker, hearer, action)
> effect: hearer WANT (hearer DO action)

The following examples show typical realizations of these speech acts in English.

> 'The train leaves at 3' intended literally is an INFORM that the train leaves at 3.
> 'Open the door' intended literally is a REQUEST that the hearer open the door.
> 'Does the train leave at 3' intended literally is a REQUEST that the hearer inform the speaker whether (INFORMIF) the train leaves at 3.

For the time being, we will assume that all speech acts are realized in their literal form, and thus can be easily identified from the input utterance.

4.3. Example 1: Providing more information than requested

This is a very simple example to give an idea of the plan inference process in operation. It has been modified from the way the actual system runs so it can be described in terms of the simple view of partial plans as one expectation and one alternative.

Let the observed action be:

REQUEST(A, S, INFORMREF(S, A, the (x : *time*):
DEPART.TIME of ⟨train1⟩ is x)
where ⟨train1⟩ = the (x : train): DEST(x,WINDSOR)

Such an action could be constructed from an utterance such as

'When does the train to Windsor leave?'

The action specifies an action cluster consisting of a REQUEST to INFORMREF that is added to the plan alternative in each partial plan. The partial plans are then examined for similarities. Within the 'BOARD' plan, A, the train to Windsor and the DEPART.TIME relation are all found to be mentioned in both the alternative and expectation (giving the plan a weight of 20). The train descriptions in the alternative and expectation are merged to form a more complete description, i.e., both the source (TORONTO) and the destination (WINDSOR) are known. With the 'MEET' plan, only A is found to be similar (giving it a weight of 5). As a consequence, the BOARD plan is strongly favored. If there were other expectations, they would be rated similarly, but most would probably have little in common with the utterance, and so would start off poorly rated. The null expectation plan starts with a token weight of 5.

After this initial processing, the partial plans are as in Fig. 1.

The initial tasks suggested are:

(1) Identify the train in the 'BOARD' plan, rated 62 (since its description was modified).

(2) Infer from the REQUEST act cluster in the BOARD plan, rated 46.

(3) Identify the trains in the 'MEET' plan, rated 25.

(4) Infer from the REQUEST act cluster in the MEET plan, rated 19.

(5) Identify the train in the NULL plan, rated 12.

(6) Infer from the REQUEST act cluster in the NULL plan, rated 9.

Identifying the train in the BOARD *plan.* Identifying the train in the BOARD plan succeeds, the assumption being made that the next train leaving is the one intended unless the speaker says otherwise. This provides further evidence that the BOARD plan is the correct one, increasing the BOARD plan's rating to 71 (weight 37) at the expense of the other partial plan ratings.

The Infer and Expand cycle. The *Infer* task on the BOARD plan (now rated 53) is executed. Inferring from (2) in the BOARD plan, the effect of the

The BOARD plan (rated 62):

The expectation:

BOARD(A,train1, TORONTO)

enable

AT(A, the x : DEPART.LOC(train1,x), the x :
DEPART.TIME(train1, x))

The alternative:

(1) REQUEST(A, S, INFORMREF(S, A, the (x : time):
DEPART.TIME(train1,x)))

effect

V

(2) S WANT INFORMREF(S, A, the (x : time): $\cdots$)

where train1 = the (x : train): SOURCE(x,TORONTO) &
DEST(x,WINDSOR)

The MEET plan (rated 25):

The expectation:

MEET(A, train2, TORONTO)

enable

AT(A, the x : ARRIVE.LOC(train2,x), the x :
ARRIVE.TIME(train2x))

where train2 = the (x : train): DEST(x, TORONTO)

The alternative:

(1) REQUEST(A, S, INFORMREF(S, A, the (x : time):
DEPART.TIME(train1,x)))

V

(2) S WANT INFORMREF(S, A, the (x : time): $\cdots$)

where train1 = the (x : train): DEST(x,WINDSOR)

The null plan (rated 12) contains only the alternative as described above in the MEET plan.

FIG. 1. The initial part plans.

REQUEST finds only the want–action rule applicable. An *Expand* task is suggested (rated 53), which immediately executes, since it is the best rated task, adding the action

(3) INFORMREF(S, A, the (x : time): $\cdots$)

Added in the action cluster with (3) is its effect

(4) A KNOWREF the (x : time): DEPART.TIME(train1,x)

and another *Infer* task is suggested from (4). Since an inference has been applied, the BOARD plan's rating is increased (heuristic (H6)) to 75 (weight 46). This *Infer* and *Expand* cycle is executed again. The know-term rule finds a link between (4) and the DEPART.TIME relation in the precondition to the BOARD action expectation. The monitor notices this intersection between the alternative and the expectation (and boosts the rating by heuristic (H5), to 82 (weight 69)), and suggests an *Accept* task (rated 82). This task terminates the plan inference for there are no other well-rated hypotheses.

S now performs the next step to help A, namely, find the obstacles in A's plan and select from them some goals to achieve. One obstacle is straightforward, A has explicitly indicated the goal of knowing the departure time of the train. However, S examines the plan further and finds the implicit obstacle that A needs to know the departure location. S accepts these goals as his own and plans to achieve each goal simply with an inform. When this plan is executed, output is produced that corresponds to the English utterances:

'The train leaves at 1600'

'The train leaves from gate 7'

An interesting problem remains as to how S could have planned to achieve both goals with a single utterance such as

'1600 at gate 7'

How does one construct a plan to achieve multiple goals simultaneously?

4.4. Example 2: A yes/no question answered no

In this section we consider the question 'Does the Windsor train leave at 4?'. The initial utterance is mapped into the action

REQUEST(A, S, INFORMIF(S, A, LEAVE(train1,1600)))
where train1 =
the (x : train): PROPERTY of x is WINDSOR.

LEAVE is a predicate pattern that will match the predicates DEPART.TIME and DEPART.LOC. The inferences from this action will eventually produce the goal (using similar steps as in above example).

A KNOWIF LEAVE(train1, 1600).

The possible knowledge-based rules suggest goals for A such as

'A wants the train to leave at 4'	[know-positive]
'A wants the train not to leave at 4'	[know-negative]
'A wants to know what train leaves at 4'	[know-value]
'A wants to know when the train leaves'	[know-value]

Only the latter goal leads to a reasonable plan; the know-term rule from it produces a connection to the third argument of the precondition to the BOARD action.

AT(A, the $(x:\text{loc})$: DEPART.LOC(train1,x),
the $(x:\text{time})$: DEPART.TIME(train1,x)).

Possible obstacles in this plan are found to be the explicitly mentioned goal of knowing whether the train leaves at 1600, plus the obstacles of knowing the departure time and location. For the sake of clarity, let us assume that the location is already known in this example. The obstacles remaining are

A KNOWIF DEPART.TIME(train1,1600)

A KNOWREF the $(x:\text{time})$: DEPART.TIME(train1,x)

If the answer to the original query were 'yes', then both these goals would be accomplished by answering the query as a yes/no question. But if the answer is 'no', only the first obstacle is achieved by the yes/no answer. The second obstacle accounts for the extra information.

This example reflects a general point. When a person asks about the truth of some proposition that happens to be false, he often is interested in a related, true proposition. The main problem is determining how to modify the original proposition to make it true. Our feeling is that, with respect to a given set of goals, the objects referred to by the terms in a proposition can usually be ordered by some criteria reflecting their importance in the plan. The more important the term, the less likely it is that it is what is wrong in the proposition. Two indicators or importance are

(1) at what level of abstraction the object (or term) was introduced into the plan, and

(2) what objects are defined in terms of other objects.

The second case occurred in the above example, the departure time was defined in terms of the train description, and so was the prime candidate to be wrong. This approach seems to be quite general. As an example, consider a 'co-operative' response cited by Kaplan [13].

(1a) 'Is John a senior?'

(1b) 'No, he's a junior.'

It makes little sense to consider an answer out of context. For instance, if I am a professor needing a senior to do some project for me, a more appropriate response to my query (1a) would be

'No, but Sam is'

This is because my goal of finding a senior is more important than John's status. If, on the other hand, my goal were to find out more about John, then the concept 'John' would be more important, hence response (1b) would be appropriate.

4.5. Example 3: An indirect speech act

Consider the plan that must be inferred in order to answer (1) with (2):

(1) A: Do you know when the Windsor train leaves?

(2) S: Yes, at 3:15.

The goal inferred from the literal interpretation is that

(3) A KNOWIF (S KNOWREF 'departure time').

Applying the know-positive rule, we obtain the goal

(4) S KNOWREF 'departure time'

which enables P to perform the action (precondition-action rule)

(5) INFORMREF (S,A, 'departure time')

to achieve the goal (action–effect rule)

(6) A KNOWREF 'departure time'

S's response (2) indicates that he believed that both (3) and (6) were obstacles that S could overcome.

However, sentences such as (1) are often uttered in a context where the literal goal is not an obstacle. For instance, A might already know the departure time, yet still utter (1). In such cases, A's goals are the same as though he had uttered the request

(7) When does the Windsor train leave?

Hence, (1) is often referred to as an *indirect speech act* [24].

Although the mechanisms already described are capable of answering such indirect acts correctly, they cannot distinguish between the two following cases:

(a) A said (1) merely expecting a yes/no answer, but S answered with the extra information in order to be helpful;

(b) A said (1) intending that S deduce his plan and realize that A really wants to know the departure time.

Theoretically, these are very different; (a) describes a yes/no question, while (b) describes an (indirect) request for the departure time. But the distinction is also important for practical reasons. For instance, assume S is not able to tell A the departure time for some reason. With interpretation (a), S can simply answer the question, whereas, with interpretation (b), S is obliged to give a reason for not answering with the departure time.

We have to reformulate our speech act definitions in order to handle such cases, as well as to bring our work in line with the philosophical views. We introduce a new set of *surface speech acts* that correspond directly to the form of the utterance. For example, an imperative mood sentence is always a surface request act (S.REQUEST) whether it is interpreted directly or not. An indicative mood sentence is always an S.INFORM act, which is defined simply as

S.INFORM(speaker, hearer, proposition)
effect: hearer BELIEVE
    speaker WANT
      hearer KNOW proposition.

The speech acts at the other level are defined by intentions of the speaker and correspond to the *illocutionary acts* in Austin [2]. These acts are executed by executing some surface act. An essential condition for the performance of an illocutionary act is that the hearer recognize that the speaker intended to perform that act. This condition could be represented as a precondition, but we represent it as a goal to achieve in the body of the act in order to allow hierarchical planning. So the definition of the illocutionary act INFORM is

INFORM(speaker, hearer, prop)
precondition: speaker KNOW prop
effect: hearer KNOW prop
body: hearer BELIEVE
    speaker WANT
      hearer KNOW prop

This is simply the old definition of INFORM with the added condition concerning the recognition of intention. The other speech act definitions are augmented similarly. Notice that the effect of the S.INFORM act matches the body of the INFORM act. This indicates that indicative sentences are a way of achieving an S.INFORM. It is important, however, that it is not the only way.

The second modification to the model is to allow the plan inference rules to be performed with the added condition that the hearer believes the speaker intended the hearer to perform it. This we do by introducing a new set of inference rules corresponding to the original ones, but with the added condition that the hearer believes the speaker intended the inference to be made.

Using these mechanisms, we say that a speaker can perform a speech act ACT by performing another speech act ACT' if he intends that the hearer recognize that ACT' was performed and also that he intended the hearer to infer that the effects of ACT should also be achieved.

Details of this analysis of speech acts can be found in [16].

5. Analyzing Sentence Fragments

As we have seen, plan knowledge is necessary to generate appropriate responses even to syntactically complete sentences. The mood of a sentence, given by the subject, auxiliaries, and main verb, is critical to speech act identification. With sentence fragments such as 'the Montreal train', even the mood of the sentence may not be known, thus making even the surface speech act identification difficult. However, the plan inference process described so far is already powerful enough to handle many ambiguities of this type.

Even in sentence fragments, there remain syntactic clues to the surface speech act. Words such as 'when', 'what', 'which', etc., signal a S.REQUEST to INFORMREF. The use of the word 'please' in a sentence marks it as a request [21]. Thus, an utterance such as 'the door, please' could not be interpreted as an inform. Of course, there will often be cases where a mood ambiguity cannot be resolved at the syntactic level, and in these cases, the alternatives will be enumerated and each case will become a plan alternative. Since the number of surface speech acts is small, this approach is reasonable.

The less explicit the utterance, the more important the expectations become, for they provide the missing details of the speaker's actions and plans. Typically, a speaker has a specific speech act and propositional content that he wants to convey to the hearer. In addition, the speaker may have some idea of what the hearer expects him to say. Any fragment that singles out the correct expectation from the rest is acceptable to communicate the speech act and proposition. The fragment must also distinguish what particular subgoals in the expectation are being pursued. In restrictive domains, such as the train station, identifying the fundamental goal (i.e. boarding, meeting) is sufficient to identify the subgoals desired. In such settings, very brief fragments can be used successfully. For example,

'The train to Windsor?'

successfully identifies the fundamental goal of boarding the train. Of the possible subgoals that are involved in this plan, only knowing the departure time and location are relevant (expected), for this is what the information agent believes that the patron believes he can help achieve. Other subgoals required in order to board the train such as having a ticket, are not relevant because (the information agent believes) the patron believes the information agent does not handle tickets.

5.1. An example of a sentence fragment

As usual, the setting is the train station, A is a patron and S is the information agent.

A: The train to Windsor?

The syntactic analysis suggests two interpretations:

(5.1) S.REQUEST(A, S, INFORMREF(S, A,
the x : PROPERTY of train1 is x))

(5.2) S.REQUEST(A, S, INFORMIF(S, A, PROPERTY involving train1))

train1 = the (x : train): PROPERTY(TO) of x is WINDSOR.

The use of PROPERTY(TO) here is a cheat; it stands for an arbitrary property (involving a train and a city in this case) that is realizable at the syntactic level using the preposition 'to'. The problem we are avoiding here is that the actual relation referred to here can only be obtained from the expectations, which are not considered until the sentence is parsed and the 'literal' meaning constructed. It is not a simple matter to change this though, for arbitrarily many inferences may have to be made from the literal meaning before the correct relation can be identified. We have resorted to encoding such syntactic restrictions in special patterns that match the appropriate relation names.

This example will consider only the first interpretation. Details on how the second is eliminated can be found in [1]. The described train is incompatible with the MEET expectation, leaving only the BOARD expectation as the reasonable interpretation. The inferences made from interpretation (5.1) lead to the goal:

A KNOWREF the x : PROPERTY of 'train' is x

To identify the actual predicate indicated by the predicate pattern PROPERTY, the BOARD expectation is inspected for matches.

There are two relevant properties of trains, the DEPART.TIME and the DEPART.LOC. Assuming that S believes that A knows neither of the values for these relations, both can be considered obstacles and be used to form a response corresponding to

'It leaves at 3:15 from gate 7.'

In another setting, S's response to the same fragment might be quite different. If the train station had only one platform, he would only respond with the departure time because he would believe that A knows the location already. To be completely different, if S were the ticket agent he would interpret the fragment as a request for a ticket (since this is what S expects, i.e. what S believes that A believes S is able to do), and might reply

'$10.50 please'

This approach covers a quite different range of sentence fragments than any other method described in the literature. The most common method, which could be called the 'semantic approach', accepts fragments in the form of full syntactic units, such as noun phrases, and uses the fragments to build a partial 'semantic' representation that is then matched into the representation of the previous utterance [9, 3, 10]. If this match is successful, the representation of the utterance is constructed out of the previous utterance's structure with the newly specified parts replacing the parts that they matched. This method is limited to those fragments that depend on the structure of the previous utterance for their interpretation. As shown in the train dialogues, there are many fragments used where this is not the case.

Our approach is suited for cases where the mere mention of a concept or phrase is suggestive enough to convey a thought/wish. These instances typically have little syntactic relation to previous utterances, and in fact can occur when there is no previous utterance. In many ways, the matching techniques are similar to the 'semantic approach', but the goals are very different. The goal of the semantic approach is to find a structural similarity with the previous utterance. The goal of this work is to identify the plan and goals of the speaker. A syntactically complete utterance is never considered or constructed for it has no effect on the understanding of the utterance.

6. Clarification Dialogues

Many dialogue situations do not follow a simple question–answer ordering. Variations often arise because of misunderstandings or ambiguity in previous utterances. A plan-based model can account for some of these variations in a straightforward manner.

Consider the following exchange between a patron and the clerk at the train station.

(6.1) When is the Windsor train?

(6.2) To Windsor?

(6.3) Yes.

(6.4) 3:15.

Why is (6.2) a reasonable response to (6.1)? According to our model, the clerk is unable to deduce the plan of the patron from (6.1). In particular, two plans seem possible:

(a) the patron wants to meet a train arriving from Windsor;

(b) the patron wants to board a train departing for Windsor.

Utterance (6.2) is intended to resolve this ambiguity by establishing a fact that will distinguish between the two plans.

To consider the model in detail, it is necessary to examine the relationship between the plan inference process and plan construction. The plan inference process can be viewed as an abstract action (the PI action), which is performed by a set of subactions (the PI tasks). Typically, the PI action is not expanded in a plan, its execution is in effect 'compiled' and runs without requiring any explicit reasoning about its execution. However, when difficulties arise, say one of the PI tasks cannot execute successfully unless some obstacle (in the sense used throughout) is overcome, the agent will want to explicitly reason about the PI process. The goal that is the obstacle is explicitly added to the plan as part of the body of the PI action. Hence, planning can be initiated on it in the standard manner, and once the goal is achieved, the 'compiled' PI execution can resume.

For example, consider a train setting in which there is a train scheduled to leave for Winsor at the same time as one is scheduled to arrive from Windsor. S is executing the PI action in an effort to help the patron who asks:

A: When is the Windsor train?

which is analyzed as

S.REQUEST(A, S, INFORMREF(S, A,
the (x : time): PROPERTY of train1 is x))
train1 = the (x : train): PROPERTY of x is WINDSOR

This action can be interpreted with respect to either the BOARD or MEET expectation. With the BOARD expectation, the train leaves TORONTO for WINDSOR and A is interested in its departure time. With the MEET expectation, the train arrives in TORONTO from WINDSOR and A is interested in its arrival time. S has no reason to favor one reading over the other, so both alternatives are investigated to a uniform extent and eventually suggestions are made for both that they be accepted as the interpretation.

The *Accept* task detects that there is an ambiguity and, after allowing the PI process to continue to try and disambiguate to no avail, creates a goal to disambiguate the utterance. For instance, assuming that the *Accept* task that involves the BOARD interpretation is selected first by the search mechanism, the goal created is

S KNOWIF (A WANT (BOARD(A, train1, TORONTO))

To achieve this goal, S creates a plan to (i) ask A whether BOARDing the train is his goal, and (ii) receive back A's response. So S executes the action

S.REQUEST(S, A,
INFORMIF(A, S, AW(BOARD(A, train1, TORONTO))

producing an utterance corresponding to

'Do you want to go to Windsor?'

and then waits for a response. The response is handled by the PI action (nested inside the body of the original PI action) with the expectation that A will execute the action

INFORMIF(A, S, AW(BOARD(A, train1, TORONTO)))

Because this expectation is so explicit, even very brief responses with little content, such as a simple 'yes' or 'no', can be understood easily.

Assuming that the answer is 'yes', S then knows that A wants to BOARD the train. This knowledge achieves the goal specified by the *Accept* task and allows it to arrive at a (successful) decision. The BOARD interpretation is accepted and the PI process terminates. The obstacle detection and the planning of a response then continue as usual, eventually causing S to execute a response corresponding to an utterance such as

'It leaves at 3:15.'

6.1. Detecting dialogue failures

Above we considered dialogues initiated by S when he could not understand A's utterance. In this section we consider the opposite case, where A initiates a dialogue because of a failure to understand on his part, or to make a correction of a misunderstanding by S. We have not examined these dialogues in detail yet, but preliminary analysis indicates that many such utterances can be detected by using what S believes about A's plan and beliefs. Consider some examples that demonstrate the types of failure that could occur after the exchange

(6.5) A: 'When is the Windsor train?'

(6.6) B: 'It leaves at 3:15.'

At this point S believes that A now knows the train leaves at 3:15, and therefore that his stated goal of knowing the departure time is accomplished. This interpretation depends on the fact that S believes A wants to board the train to Windsor (rather than meet a train from Windsor).

A can indicate a failure by stating a goal that S believes has already been achieved in SBAW. For example, the continuation

(6.7) A: 'When?'

indicates that the goal of knowing the departure time is not yet accomplished.

To model how S could recover from such a situation we would require a detailed analysis of the possible reasons for failure, e.g. faulty communication, disbelief of answer, etc.

Other utterances indicate that S made an error in inferring A's plan. These involve A denying that he wants a goal in the inferred plan. Some examples are

(6.8) A: 'I don't want to travel to Windsor.'

(6.9) A: 'No, I want to meet the Windsor train!'

These contradict the fundamental goals that S inferred for A and indicate that S should re-analyse the previous utterance after making appropriate modifications to his expectations about A's plan. Utterance (6.9) presents a difficulty: S must be able to realize what goals are mutually exclusive. In the train domain, the fundamental goals are assumed to be mutually exclusive, but in a more complex domain, such knowledge is non-trivial.

7. Future Directions

We have argued that much linguistic behavior can be explained by assuming a plan based model of language. Our specification of the actual plan inference process, however, is not detailed enough to allow it to perform in more complex domains than the train station. Considerable work needs to be done to specify more control heuristics. Large domains probably require the introduction of domain specific inference rules. We have begun to lay the groundwork by specifying characteristics that any plan inference mechanism would need in any domain.

Along with more complex domains will come more complex dialogues; in particular, dialogues that continue for prolonged periods of time and cover a wide range of topics. We have hinted in the section on clarification dialogues that some dialogue structure can be explained by the mechanisms of the plan based model itself. It will be interesting to see how much dialogue structure can be accounted for this way, and how much must be specifically introduced as 'conventional rules'.

One of the major problems in larger domains is the effective management of the large number of potential expectations. The progress of a dialogue relies on old expectations, but more importantly, establishes new ones. Grosz [9] shows how task structure can limit these expectations. Topic shifts and the initial dialogue expectations remain unaccounted for.

Perhaps the most difficult problems lie in specifying the relation between the syntactic processing and the rest of the system. We saw in the section on sentence fragments a case where the syntactic information concerning the preposition 'to' could not be used until the advanced stages of the plan inference process. The major issue here concerns what we were working with before the relation indicated by the phrase 'the train to Windsor' was identified.

One final concern is with the fundamental tools of our approach: our logics of belief, want and action are only minimally adequate. This has been acceptable so far, for our emphasis has been on demonstrating the usefulness of such notions in a model of language. Now that we have a better idea of how these tools can be used, it is time to return to them and attempt a better formulation.

8. Conclusions

The plan based model of language described here can be used to explain a wide range of linguistic behavior that has been problematic to previous approaches. In particular, we have addressed the problem of:

— generating responses that convey more information than was explicitly requested;
— generating appropriate responses to utterances that consist solely of a sentence fragment;
— generating clarification subdialogues when the intent of a previous utterance is not clear.

The common thread through the solutions to these problems and to the understanding of indirect speech acts is the inferring of the speaker's plans and the detection of obstacles in these plans.

We have explicitly indicated the role that context plays in language understanding: only those plans that are reasonable in the current context (as determined by the rating heuristics) are potential analyses of the intention of the speaker. A large part of the context is the hearer's model of the speaker's beliefs and goals. If the context is sufficiently restrictive to uniquely determine the speaker's plan, then appropriate responses can be generated for a wide range of utterances often considered problematic. If the context is not sufficiently restrictive, the model generates goals that lead to clarification dialogues.

REFERENCES

1. Allen, J.F., A plan-based approach to speech act recognition, Thesis, Department of Computer Science, University of Toronto (1979).
2. Austin, J.L., *How To Do Things With Words* (Oxford University Press, New York, 1962).
3. Burton, R.R., Semantic grammar: An engineering technique for constructing natural language understanding systems, BBN Rep. 3453 (1976).
4. Bruce, B.C., Plans and social action, in: R. Spiro, B. Bruce and W. Brewer, Eds., *Theoretical Issues in Reading Comprehension* (Lawrence Erlbaum, Hillsdale, NJ, to appear).
5. Cohen, P.R., On knowing what to say: Planning speech acts, Techn. Rep. 118, Department of Computer Science, University of Toronto (January 1978).
6. Cole, P. and Morgan, J.L., Eds., *Syntax and Semantics, Vol. 3: Speech Acts.* (Academic Press, New York, 1975).
7. Ernst, G. and Newell, A., *GPS: A Case Study in Generality and Problem Solving* (Academic Press, New York, 1969).
8. Fikes, R.E. and Nilsson, N.J. STRIPS: A new approach to the application of theorem proving to problem solving, *Artificial Intelligence* 2 (1971) 189–205.

9. Grosz, B.G., The representation and use of focus in dialog understanding, Dissertation, University of California at Berkeley (1977).
10. Hendrix, G., Human engineering for applied natural language processing, in: *Proceedings of the 5th International Joint Conference on Artificial Intelligence*, Cambridge, MA (1977).
11. Hintikka, J., *Knowledge and Belief* (Cornell University Press, Ithaca, NY, 1963).
12. Horrigan, M.K., Modelling Simple Dialogs, TR 108, Department of Computer Science, University of Toronto (1977).
13. Kaplan, S.J., Indirect responses to loaded questions, in: *Proceedings of the 2nd Conference on Theoretical Issues in Natural Language Processing*, Champaign-Urbana (1978).
14. Lehnert, W., Human and computational question answering, *Cognitive Sci.* **1** (1) (1977).
15. Nilsson, N.J., Problem solving methods in artificial intelligence (McGraw-Hill, New York, 1971).
16. Perrault, C.R. and Allen, J.F., *A Plan-Based Analysis of Indirect Speech Acts, American Journal of Computational Linguistics*, to appear.
17. Perrault, C.R. Allen, J.F. and Cohen, P.R., Speech acts as a basis for understanding dialogue coherence, in: *Proceedings of 2nd Conference on Theoretical Issues in Natural Language Processing*, Champaign-Urbana (1978).
18. Perrault, C.R. and Cohen, P.R., Inaccurate reference, in: A.K. Joshi, I.A. Sag and B.L. Webber, Eds., *Computational Aspects of Linguistic Structure and Dialogue Setting* (Cambridge University Press, New York, to appear).
19. Sacerdoti, E.D., Planning in a hierarchy of abstraction spaces, in: *Proceedings of 3rd International Joint Conference on Artificial Intelligence*, Stanford, CA (1973).
20. Sacerdoti, E.D., The nonlinear nature of plans, in: *Proceedings of 4th International Joint Conference on Artificial Intelligence*, Tbilisi, USSR (1975).
21. Sadock, J.M., *Toward a Linguistic Theory of Speech Acts* (Academic Press, New York, 1974).
22. Schiffer, S.R., *Meaning* (Oxford University Press, London, 1972).
23. Schmidt, C.F., Sridharan, N.S. and Goodson, J.L., The plan recognition problem: An intersection of artificial intelligence and psychology, *Artificial Intelligence* **10** (1) (1979).
24. Searle, J.R., Indirect speech acts, in: P. Cole and J.L. Morgan, Eds., *Syntax and Semantics, Vol. 3: Speech Acts* (Academic Press, New York, 1975).
25. Wilensky, R., Understanding goal-based stories, Thesis, Yale University (1978).

Received 9 November 1979; revised version received 7 May 1980