Collaborating on Referring Expressions*

Peter A. Heeman
Department of Computer Science
University of Rochester†
Rochester, New York
14627
heeman@cs.rochester.edu

Graeme Hirst
Department of Computer Science
University of Toronto
Toronto, Canada
M5S 1A4
gh@cs.toronto.edu

Technical Report 435
Department of Computer Science
University of Rochester

Revised April 1995

Abstract

This paper presents a computational model of how conversational participants collaborate in order to make a referring action successful. The model is based on the view of language as goal-directed behavior. We propose that the content of a referring expression can be accounted for by the planning paradigm. Not only does this approach allow the processes of building referring expressions and identifying their referents to be captured by plan construction and plan inference, it also allows us to account for how participants clarify a referring expression by using meta-actions that reason about and manipulate the plan derivation that corresponds to the referring expression. To account for how clarification goals arise and how inferred clarification plans affect the agent, we propose that the agents are in a certain state of mind, and that this state includes an intention to achieve the goal of referring and a plan that the agents are currently considering. It is this mental state that sanctions the adoption of goals and the acceptance of inferred plans, and so acts as a link between understanding and generation.

* To appear in *Computational Linguistics*, Volume 21-3, 1995

† This research was began at the Department of Computer Science, University of Toronto, in the first author's MSc thesis under the supervision of the second author.

1 Introduction

People are goal oriented and can plan courses of actions to achieve their goals. But sometimes they might lack the knowledge needed to formulate a plan of action, or some of the actions that they plan might depend on coordinating their activity with other agents. How do they cope? One way is to work together, or *collaborate*, in formulating a plan of action with other people who are involved in the actions or who know the relevant information.

Even in the apparently simple linguistic task of referring, in an utterance, to some object or idea can involve exactly this kind of activity: a collaboration between the speaker and the hearer. The speaker has the goal of the hearer identifying the object that the speaker has in mind. The speaker attempts to achieve this goal by constructing a description of the object that she thinks will enable the hearer to identify it. But since the speaker and the hearer will inevitably have different beliefs about the world, the hearer might not be able to identify the object. Often, when the hearer cannot do so, the speaker and hearer collaborate in making a new referring expression that accomplishes the goal.

This paper presents a computational model of how a conversational participant collaborates in making a referring action successful. We use as our basis the model proposed by Clark and Wilkes-Gibbs (1986), which gives a descriptive account of the conversational moves that participants make when collaborating upon a referring expression. We cast their work into a model based on the planning paradigm.

We propose that referring expressions can be represented by plan derivations, and that plan construction and plan inference can be used to generate and understand them. Not only does this approach allow the processes of *building* referring expressions and *identifying* their referents to be captured in the planning paradigm, it also allows us to use the planning paradigm to account for how participants *clarify* a referring expression. In this case, we use meta-actions that encode how a plan derivation corresponding to a referring expression can be reasoned about and manipulated.

To complete the picture, we also need to account for the fact that the conversants are *collaborating*. We propose that the agents are in a mental state that includes not only an intention to achieve the goal of the collaborative activity but also a plan that the participants are currently considering. In the case of referring, this will be the plan derivation that corresponds to the referring expression. This plan is in the common ground of the participants, and we propose rules that are sanctioned by the mental state both for *accepting* plans that clarify the current plan, and for *adopting* goals to do likewise. The acceptance of a clarification results in the current plan being updated. So, it is these rules that specify how plan inference and plan construction affect and are affected by the mental state of the agent. Thus, the mental state, together with the rules, provides the link between these two processes. An important consequence of our proposal is that the current plan need not allow the successful achievement of the goal. Likewise, the clarifications that agents propose need not result in a successful plan in order for them to be accepted.

As can be seen, our approach consists of two tiers. The first tier is the planning component, which accounts for how utterances are both understood and generated. Using the planning paradigm has several advantages: it allows both tasks to be captured in a single paradigm that is used for modeling general intelligent behavior; it allows more of the content of an utterance to be accounted for by a uniform process; and only a single knowledge source for referring expressions is needed instead of having this knowledge embedded in special algorithms for each task. The second tier accounts for the collaborative behavior of the agents: how they adopt goals and coordinate their activity. It provides the link between the mental state of the agent and the planning processes.

In accounting for how agents collaborate in making a referring action, our work aims to make the following contributions to the field. First, although much work has been done on how agents request clarifications, or respond to such requests, little attention has been paid to the collaborative aspects of clarification discourse. Our work attempts a plan-based formalization of what linguistic collaboration is, both in terms of the goals and intentions that underlie it and the surface speech acts that result from it. Second, we address the act of referring and show how it can be better accounted for by the planning paradigm. Third, previous plan-based linguistic research has concentrated on either construction or understanding of utterances, but not both. By doing both, we will give our work generality in the direction of a complete model of the collaborative process. Finally, by using Clark and Wilkes-Gibbs's model as a basis for our work, we aim not only to add support to their model, but gain a much richer understanding of the subject.

In order to address the problem that we have set out, we have limited the scope of our work. First, we look at referring expressions in isolation, rather than as part of a larger speech act. Second, we assume that agents have mutual knowledge of the mechanisms of referring expressions and collaboration. Third, we deal with objects that both the speaker and hearer know of, though they might have different beliefs about what propositions hold for these objects. Fourth, as the input and the output to our system, we use representations of surface speech actions, not natural language strings. Finally, although belief revision is an important part of how agents collaborate, we do not explicitly address this.

2 Referring as a Collaborative Process

Clark and Wilkes-Gibbs (1986) investigated how participants in a conversation collaborate in making a referring action successful. They conducted experiments in which participants had to refer to objects—tangram patterns—that are difficult to describe. They found that typically the participant trying to describe a tangram pattern would present an initial referring expression. The other participant would then pass judgment on it, either *accepting* it, *rejecting* it, or *postponing* his decision. If it was rejected or the decision postponed, then one participant or the other would *refashion* the referring expression. This would take the form of either repairing the expression by correcting speech errors, *expanding* it by adding further qualifications, or *replacing* the original expression with a new expression. The referring expression that results from this is then judged, and the process continues until the referring expression is acceptable enough to the participants for current purposes. This final expression is contributed to the participants' common ground.

Below are two excerpts from Clark and Wilkes-Gibbs's experiments that illustrate the acceptance process.

(2.1) **A:** 1 Um, third one is the guy reading with, holding his book to the left.

B: 2 Okay, kind of standing up?

A: 3 Yeah.

B: 4 Okay.

In this dialogue, person A makes an initial presentation in line 1. Person B postpones his decision in line 2 by voicing a *tentative "okay"*, and then proceeds to refashion the referring expression, the result being "the guy reading, holding his book to the left, kind of standing up." A accepts the new expression in line 3, and B signals his acceptance in line 4.

(2.2) **A:** 1 Okay, and the next one is the person that looks like they're carrying something and it's sticking out to the left. It looks like a hat that's upside down.

B: 2 The guy that's pointing to the left again?

A: 3 Yeah, pointing to the left, that's it! (laughs)

B: 4 Okay.

In the second dialogue, B implicitly rejects A's initial presentation by replacing it with a new referring expression in line 2, "the guy that's pointing to the left again." A then accepts the refashioned referring expression in line 3.

Below, we give an algorithmic interpretation of Clark and Wilkes-Gibbs's collaborative model, where **present**, **judge**, and **refashion** are the conversational moves that the participants make, and *ref*, *re*, and *judgment* are variables that represent the referent, the current referring expression, and its judgment, respectively. (Since the conversational moves update the referring expression and its judgment, they are presented as functions.)

re = **present(***ref***)**
$judgment$ = **judge(***ref,re***)**
while (judgment ≠ **accept***)*
  re = **refashion(***ref,re***)**
  $judgment$ = **judge(***ref,re***)**
end-while

The algorithm illustrates how the collaborative activity progresses by the participants judging and refashioning the previously proposed referring expression.[1] In fact, we can see that the *state* of the process is characterized by the current referring expression, *re*, and the judgment of it, *judgment*, and that this state must be part of the common ground of the participants. The algorithm also illustrates how the model of Clark and Wilkes-Gibbs minimizes the distinction between the roles of the person who initiated the referring expression and the person who is trying to identify it. Both have the same moves available to them, for either can judge the description and either can refashion it. Neither is controlling the dialogue, they are simply collaborating.

In later work, Clark and Schaefer (1989) propose that "each part of the acceptance phase is itself a contribution" (p. 269), and the acceptance of these contributions depends on whether the hearer "believes he is understanding well enough for current purposes" (p. 267). Although Clark and Schaefer use the term *contribution* with respect to the discourse, rather than the collaborative effort of referring, their proposal is still relevant here: judgments and refashionings are contributions to the collaborative effort and are subjected to an acceptance process, with the result being that once they are accepted, the state of the collaborative activity is updated. So, what constitutes grounds for accepting a judgment or clarification? In order to be consistent with Clark and Wilkes-Gibbs' model, we can see that if one agent finds the current referring expression problematic, the other must accept that judgment. Likewise, if one agent proposes a referring expression, through a refashioning, the other must accept the refashioning.

3 Referring Expressions

3.1 Planning and Referring

By viewing language as action, the planning paradigm can be applied to natural language processing. The actions in this case are *speech acts* (Austin, 1962; Searle, 1969), and include such things as promising, informing, and requesting. Cohen and Perrault (1979) developed a system that uses plan construction to map an agent's goals to speech acts, and Allen and Perrault (1980) use plan inference to understand an agent's plan from its speech acts. By viewing it as action (Searle, 1969), referring can be incorporated into a planning model. Cohen's model (1981) planned requests that the hearer identify a referent, whereas Appelt (1985) planned *concept activations*, a generalization of referring actions.

Although acts of reference have been incorporated into plan-based models, determining the content of referring expressions hasn't been. For instance, in Appelt's model, concept activations can be achieved by the action *describe*, which is a primitive, not further decomposed. Rather, this action has an associated procedure that determines a description that satisfies the preconditions of *describe*. Such special procedures have been the mainstay for accounting for the content of referring expressions, both in constructing and in understanding them, as exemplified by Dale (1989), who chose descriptors on the basis of their discriminatory power, Ehud Reiter (1990), who focused on avoiding misleading conversational implicatures when generating descriptions, and Mellish (1985), who used a constraint satisfaction algorithm to identify referents.

Our work follows the plan-based approach to language generation and understanding. We

[1]For simplicity, we have not shown the change in speakers between refashionings and judgments.

extend the earlier approaches of Cohen and Appelt by accounting for the content of the description at the planning level. This is done by having surface speech actions for each component of a description, plus a surface speech action that expresses a speaker's intention to refer. A referring action is composed of these primitive actions, and the speaker utters them in her attempt to refer to an object.

These speech actions are the building blocks that referring expressions are made from. Acting as the mortar are intermediate actions, which have constraints that the plan construction and plan inference processes can reason about. These constraints encode the knowledge of how a description can allow a hearer to identify an object. First, the constraints express the conditions under which an attribute can be used to refer to an object; for instance, that it be mutually believed that the object has a certain property (Clark and Marshall, 1981; Perrault and Cohen, 1981; Nadathur and Joshi, 1983). Second, the constraints keep track of which objects could be believed to be the referent of the referring expression. Third, the constraints ensure that a sufficient number of surface speech actions are added so that the set of candidates associated with the entire referring expression consists of only a single object, the referent. These constraints enable the speaker to construct a referring expression that she believes will allow the hearer to identify the referent. As for the hearer, the explicit encoding of the adequacy of referring expressions allows referent identification to fall out of the plan inference process.

Our approach to treating referring as a plan in which surface speech actions correspond to the components of the description allows us to capture how participants collaborate in building a referring expression. Plan repair techniques can be used to refashion an expression if it is not adequate, and clarifications can refer to the part of the plan derivation that is in question or is being repaired. Thus we can model a collaborative dialogue in terms of the changes that are being made to the plan derivation.

The referring expression plans that we propose are not simply data structures, but are mental objects that agents have beliefs about (Pollack, 1990). The plan derivation expresses beliefs of the speaker: how actions contribute to the achievement of the goal, and what constraints hold that will allow successful identification.[2] So plan construction reasons about the beliefs of the agent in constructing a referring plan; likewise, plan inference, after hypothesizing a plan that is consistent with the observed actions, reasons about the other participant's (believed) beliefs in satisfying the constraints of the plan. If the hearer is able to satisfy the constraints, then he will have understood the plan and be able to identify the referent, since a term corresponding to it would have been instantiated in the inferred plan. Otherwise, there will be an action that includes a constraint that is unsatisfiable, and the hearer construes the action as being in error. (We do not reason about how the error affects the satisfiability of the goal of the plan nor use the error to revise the beliefs of the hearer.)

3.2 Vocabulary and Notation

Before we present the action schemas for referring expressions, we need to introduce the notation that we use. Our terminology for planning follows the general literature.[3] We use the terms *action schema*, *plan derivation*, *plan construction*, and *plan inference*. An action schema consists of a *header*, *constraints*, a *decomposition*, and an *effect*; and it encodes the constraints under which an effect can be achieved by performing the steps in the decomposition. A plan derivation is an instance of an action that has been recursively expanded into primitive actions—its *yield*. Each component in the plan—the action headers, constraints, steps, and effects—are referred to as *nodes* of the plan, and are given names so as to distinguish two nodes that have the same content. Finally, plan construction is the process of finding a plan derivation whose yield will achieve a given effect, and plan inference is the process of finding a plan derivation whose yield is a set of observed primitive actions.

The action schemas make use of a number of predicates and these are defined in Table 1. We adopt the Prolog convention that variables begin with an upper-case letter, and all predicates and

[2] Since we assume that the agents have mutual knowledge of the action schemas and that agents can execute surface speech actions, we do not consider beliefs about generation or about the executability of primitive actions.

[3] See the introductory chapter of Allen, Hendler, and Tate (1990) for an overview of planning.

constants begin with a lower-case letter. Two constants that need to be mentioned are *system* and *user*. The first denotes the agent that we are modeling, and the latter, her conversational partner. Since the action schemas are used for both constructing the plans of the *system*, and inferring the plans of the *user*, it is sometimes necessary to refer to the speaker or hearer in a general way. For this we use the propositions *speaker(Speaker)* and *hearer(Hearer)*. These instantiate the variables *Speaker* and *Hearer* to *system* or *user*; which is which depends on whether the rule is being used for plan construction or plan inference. These propositions are included as constraints in the action schemas as needed.

3.3 Action Schemas

This section presents action schemas for referring expressions. (We omit discussion of actions that account for superlative adjectives, such as "largest", that describe an object relative to the set of objects that match the rest of the description. A full presentation is given by Heeman (1991).)

As we mentioned, the action for referring, called *refer*, is mapped to the surface speech actions through the use of intermediate actions and plan decomposition. All of the reasoning is done in the *refer* action and the intermediate actions, so no constraints or effects are included in the surface speech actions.

We use three surface speech actions. The first is *s-refer(Entity)*, which is used to express the speaker's intention to refer. The second is *s-attrib(Entity,Predicate)*, and is used for describing an object in terms of an attribute; *Entity* is the discourse entity of the object, and *Predicate* is a lambda expression, such as $\lambda X \cdot$ *category(X,bird)*, that encodes the attribute. The third is *s-attrib-rel(Entity,OtherEntity,Predicate)*, and is used for describing an object in terms of some other object. In this case *Predicate* is a lambda expression of two variables, one corresponding to *Entity*, and the other to *OtherEntity*; for instance, $\lambda X \cdot \lambda Y \cdot$ *in(X,Y)*.

Refer Action

The schema for *refer* is shown in Figure 1. The *refer* action decomposes into two steps: *s-refer*, which expresses the speaker's intention to refer, and *describe*, which accounts for the content of the referring expression (given next). The effect of *refer* is that the hearer should believe that the speaker has a goal of the hearer knowing the referent of the referring expression. The effect has been formulated in this way because we are assuming that when a speaker has a communicative goal she plans to achieve the goal by making the hearer recognize it; the effect will be achieved by the hearer inferring the speaker's plan, regardless of whether or not the hearer is able to determine the actual referent. To simplify our implementation, this is the only effect that is stated for the action schemas for referring expressions. It corresponds to the literal goal that Appelt and Kronfeld (1987) propose (whereas the actual identification is their condition of satisfaction).

Intermediate Actions

The *describe* action, shown in Figure 2, is used to construct a description of the object through its decomposition into *headnoun* and *modifiers*. The variable *Cand* is the candidate set, the set of

Header:	*refer(Entity,Object)*
Constraint:	*knowref(Speaker,Speaker,Entity,Object)*
Decomposition:	*s-refer(Entity)*
	describe(Entity,Object)
Effect:	*bel(Hearer,goal(Speaker,knowref(Hearer,Speaker,Entity,Object)))*

Figure 1: *refer* schema

Belief

*bel(Agt,Prop)***:** *Agt* believes that *Prop* is true.

*bmb(Agt1,Agt2,Prop)***:** *Agt1* believes that it is mutually believed between himself and *Agt2* that *Prop* is true.

*knowref(Agt1,Agt2,Ent,Obj)***:** *Agt1* knows the referent that *Agt2* associates with the discourse entity *Ent* (Webber, 1983), which *Agt1* believes to be *Obj*. (Proving this proposition with *Ent* unbound will cause a unique identifier to be created for *Ent*.)

Goals and Plans

*goal(Agt,Goal)***:** *Agt* has the goal *Goal*. Agents act to make their goals true.

*plan(Agt,Plan,Goal)***:** *Agt* has the goal of *Goal* and has adopted the plan derivation *Plan* as a means to achieve it. The agent believes that each action of *Plan* contributes to the goal, but not necessarily that all of the constraints hold; in order words, the plan must be coherent but not necessarily valid (Pollack, 1990, p. 94).

*content(Plan,Node,Content)***:** The node named by *Node* in *Plan* has content *Content*.

*yield(Plan,Node,Actions)***:** The subplan rooted at *Node* in *Plan* has a yield of the primitive actions *Actions*.

*achieve(Plan,Goal)***:** Executing *Plan* will cause *Goal* to be true.

*error(Plan,Node)***:** *Plan* has an error at the action labeled *Node*. Errors are attributed to the action that contains the failed constraint. This predicate is used to encode an agent's belief about an invalidity in a plan.

Plan Repair

*substitute(Plan,Node,NewAction,NewPlan)***:** Undo all variable bindings in *Plan* (except those in primitive actions, and then substitute the action header *NewAction* into *Plan* at *Node*, resulting in the partial plan *NewPlan*.

*replan(Plan,Actions)***:** Complete the partial plan *Plan*. *Actions* are the primitive actions that are added to the plan.

*replace(Plan,NewPlan)***:** The plan *NewPlan* replaces *Plan*.

Miscellaneous

*subset(Set,Lambda,Subset)***:** Compute the subset, *Subset*, of *Set* that satisfies the lambda expression *Lambda*.

*modifier-absolute-pred(Pred)***:** *Pred* is a predicate that an object can be described in terms of. Used by the *modifier-absolute* schema given in Figure 6.

*modifier-relative-pred(Pred)***:** *Pred* is a predicate that describes the relationship between two objects. Used by the *modifier-relative* schema given in Figure 7.

*pick-one(Object,Set)***:** Pick one object, *Object*, of the members of *Set*.

*speaker(Agt)***:** *Agt* is the current speaker.

*hearer(Agt)***:** *Agt* is the current hearer.

Table 1: Predicates and Actions

potential referents, associated with the head noun that is chosen, and it is passed to the *modifiers* action so that it can ensure that the rest of the description rules out all of the alternatives.

Header:	*describe(Entity,Object)*
Decomposition:	*headnoun(Entity,Object,Cand)*
	modifiers(Entity,Object,Cand)

Figure 2: *describe* schema

The action *headnoun*, shown in Figure 3, has a single step, *s-attrib*, which is the surface speech action used to describe an object in terms of some predicate, which for the *headnoun* schema, is restricted to the category of the object.[4] The schema also has two constraints. The first ensures that the referent is of the chosen category and the second determines the candidate set, *Cand*, associated with the head noun that is chosen. The candidate set is computed by finding the subset of the objects in the world that the speaker believes could be referred to by the head noun—the objects that the speaker and hearer have an appropriate mutual belief about.

Header:	*headnoun(Entity,Object,Cand)*
Constraint:	*world(World)*
	bmb(Speaker,Hearer,category(Object,Category))
	subset(World,$\lambda X\cdot$ *bmb(Speaker,Hearer,category(X,Category)),Cand)*
Decomposition:	*s-attrib(Entity,*$\lambda X\cdot$ *category(X,Category))*

Figure 3: *headnoun* schema

The *modifiers* action attempts to ensure that the referring expression that is being constructed is believed by the speaker to allow the hearer to uniquely identify the referent. We have defined *modifiers* as a recursive action, with two schemas.[5] The first schema, shown in Figure 4, is used to terminate the recursion, and its constraint specifies that only one object can be in the candidate set.[6] The second schema, shown in Figure 5, embodies the recursion. It uses the *modifier* plan, which adds a component to the description and updates the candidate set by computing the subset of it that satisfies the new component. The *modifier* plan thus accounts for individual components of the description.

Header:	*modifiers-terminate(Entity,Object,Cand)*
Constraint:	*Cand = [Object]*
Decomposition:	*null*

Figure 4: *modifiers* schema for terminating the recursion

There are two different action schemas for *modifier*; one is for absolute modifiers, such as "black" and the other is for relative modifiers, such as "larger". The former is shown in Figure 6; it decomposes into the surface speech action *s-attrib* and has a constraint that determines the new candidate set, *NewCand*, by including only the objects from the old candidate set, *Cand*, for which the predicate could be believed to be true. The other schema is shown in Figure 7 and is used for describing objects in terms of some other object. It uses the surface speech action *s-attrib-rel* and also includes a step to refer to the object of comparison.

[4]Note that several category predications might be true of an object, and we do not explore which would be best to use, but see Edmonds (1994) for how preferences can be encoded.

[5]We use specialization axioms (Kautz and Allen, 1986) to map the *modifiers* action to the two schemas: *modifiers-terminate* and *modifiers-recurse*.

[6]In order to distinguish this action from the primitive actions, it has a step that is marked *null*.

Header:	*modifiers-recurse(Entity,Object,Cand)*
Decomposition:	*modifier(Entity,Object,Cand,NewCand)*
	modifiers(Entity,Object,NewCand)

Figure 5: *modifiers* schema for recursing

Header:	*modifier-absolute(Entity,Object,Cand,NewCand)*
Constraint:	*modifier-pred(Pred)*
	bmb(Speaker,Hearer,Pred(X))
	subset(Cand,λX· bmb(Speaker,Hearer,Pred(X)),NewCand)
Decomposition:	*s-attrib(Entity,Pred)*

Figure 6: *modifier* schema for absolute modifiers

3.4 Plan Construction and Plan Inference

The goals that we are interested in achieving are communicative goals. Since these goals cannot be directly achieved by a plan of action, the speaker must instead plan actions that will achieve them indirectly, for instance by planning an utterance that results in the hearer recognizing her goal. So, if the speaker wants to achieve *Goal*, she will attempt to construct a plan whose effect is *bel(Hearer,goal(Speaker,Goal))*.

Plan Construction

Given an effect, the plan constructor finds a plan derivation that has a minimal number of primitive action, that is valid (with respect to the planning agent's beliefs), and whose root action achieves the effect. The plan constructor uses a best-first search strategy, expanding the derivation with the fewest number of surface speech actions. The yield of this plan derivation can then be given as input to a module that generates the surface form of the utterance.

Plan Inference

Following Pollack (1990), our plan inference process can infer plans in which, in the hearer's view, a constraint does not hold. In inferring a plan derivation, we first find the set of plan derivations that account for the primitive actions that were observed, without regard to whether the constraints hold. This is done by using a chart parser that parses actions rather than words (Sidner, 1985; Vilain, 1990). For referring plans that contain more than one modifier, there will be multiple derivations corresponding to the order of the modifiers. We avoid this ambiguity by choosing an arbitrary ordering of the modifiers for each such plan.

In the second part of the plan inference process, we evaluate each derivation by attempting to find an instantiation for the variables such that all of the constraints hold with respect to the hearer's

Header:	*modifier-relative(Entity,Object,Cand,NewCand)*
Constraint:	*modifier-rel-pred(Pred)*
	bmb(Speaker,Hearer,Pred(Object,OtherObject))
	subset(Cand,λX· bmb(Speaker,Hearer,Pred(X)(Other)),NewCand) v
Decomposition:	*s-attrib-rel(Entity,OtherEntity,Pred)*
	refer(OtherEntity,Other)

Figure 7: *modifier* schema for relative modifiers

beliefs about the speaker's beliefs. It could however be the case that there is no instantiation, either because this is not the right derivation or because the plan is based on beliefs not shared by the speaker and the hearer. In the latter case, we need to determine which action in the plan is to blame, so that this knowledge can be shared with the other participant.

After each derivation has been evaluated, if there is just one valid derivation, an instantiated derivation whose constraints all hold, then the hearer will believe that he has understood. If there is just one derivation and it is invalid, the action containing the constraint that is the source of the invalidity is noted. (We have not explored ambiguous situations, those in which more than one valid derivation remains, or, in the absence of validity, more than one invalid derivation.)

We now need to address how we evaluate a derivation. In the case where the plan is invalid, we need to partially evaluate the plan in order to determine which action contains a constraint that cannot be satisfied. However, any instantiation will lead to some constraint being found not to hold. Care must therefore be taken in finding the right instantiation so that blame is attributed to the action at fault. So, we evaluate the constraints in order of mention in the derivation, but postpone any constraints that have multiple solutions until the end. We have found that this simple approach can find the instantiation for valid plans and can find the action that is in error for the others.

To illustrate this, consider the *headnoun* action, which has the following constraints.

speaker(Speaker)
hearer(Hearer)
world(World)
bmb(Speaker,Header,category(Object,Category))
subset(World,$\lambda X\cdot$ bmb(Speaker,Hearer,category(X,Category)),Cand)

During the first step, finding the derivation, all co-referential variables will be unified. In particular, the variable *Category* will be instantiated from the co-referential variable in the surface speech action. The first three constraints have only a single solution, so they are instantiated. The fourth constraint contains *Object*. If there is exactly one object that the system believes to be mutually believed to be of *Category*, then *Object* is instantiated to it. If there is none, then this constraint is unsatisfiable, and so the evaluation of this plan stops with this action marked as being in error, since no object matches this part of the description. If there is more than one, then this constraint is postponed and the evaluator moves on to the *subset* constraint. This constraint has one uninstantiated variable, *Cand*, which has a unique (non-null) solution, namely the candidate set associated with the head noun. So, this constraint is evaluated.

The evaluation then proceeds through the actions in the rest of the plan. Assuming that no intervening errors are encountered, the evaluator will eventually reach the constraint on the terminating instance of *modifiers*, *Cand = [Object]*, with *Cand* instantiated to a non-null set. If *Cand* contains more than one object, then this constraint will fail, pinning the blame on the terminating instance of *modifiers* for there not being enough descriptors to allow the referent to be identified. Otherwise, the terminating constraint will be satisfiable, and so *Object* will be instantiated to the single object in the candidate set. This will then allow all of the mutual belief constraints that were postponed to be evaluated, since they will now have only a single solution.

4 Clarifications

4.1 Planning and Clarifying

Clark and Wilkes-Gibbs (1986) have presented a model of how conversational participants collaborate in making a referring action successful (see section 2 above). Their model consists of conversational moves that express a judgment of a referring expression and conversational moves that refashion an expression. However, their model is not computational. They do not account for how the judgment is made, how the judgment affects the refashioning, nor the content of the moves.

Following the work of Litman and Allen (1987) in understanding clarification subdialogues, we formalize the conversational moves of Clark and Wilkes-Gibbs as discourse actions. These discourse actions are meta-actions that take as a parameter a referring expression plan. The constraints and decompositions of the discourse actions encode the conditions under which they can be applied, how the referring expression derivations can be refashioned, and how the speaker's beliefs can be communicated to the hearer. So, the conversational moves, or clarifications, can be generated and understood within the planning paradigm.[7]

Surface Speech Actions

An important part of our model is the surface speech actions. These actions serve as the basis for communication between the two agents, and so they must convey the information that is dictated by Clark and Wilkes-Gibbs's model. For the judgment plans, we have the surface speech actions *s-accept*, *s-reject*, and *s-postpone* corresponding to the three possibilities in their model. These take as a parameter the plan that is being judged, and for *s-reject*, also a subset of the speech actions of the referring expression plan. The purpose of this subset is to inform the hearer of the surface speech actions that the speaker found problematic. So, if the referring expression was "the weird creature", and the hearer couldn't identify anything that he thought "weird", he might say "what weird thing", thus indicating he had problems with the surface speech action corresponding to "weird".

For the refashioning plans, we propose that there is a single surface speech action, *s-actions*, that is used for both replacing a part of a plan, and expanding it. This action takes as a parameter the plan that is being refashioned, and a set of surface speech actions that the speaker wants to incorporate into the referring expression plan. Since there is only one action, if is in uttered in isolation, it will be ambiguous between a replacement and an expansion; however, the speech action resulting from the judgment will provide the proper context to disambiguate its meaning. In fact, during linguistic realization, if the two actions are being uttered by the same person, they could be combined into a single utterance. For instance, the utterance "no, the red one" could be interpreted as a *s-reject* of the color that was previously used to describe something and an *s-actions* for the color "red."

So, as we can see, the surface speech actions for clarifications operate on components of the plan that is being built, namely the surface speech actions of referring expression plans. This is consistent with our use of plan derivations to represent utterances. Although we could have viewed the clarification speech actions as acts of informing (Litman and Allen, 1987), this would have shifted the complexity into the parameter of the *inform* and it is unclear whether anything would have been gained. Instead, we feel that a parser with a model of the discourse and the context can determine the surface speech actions.[8] Additionally, it should be easier for the generator to determine an appropriate surface form.

Judgment Plans

The evaluation of the referring expression plan indicates whether the referring action was successful or not. If it was successful, then the referent has been identified, and so a goal to communicate this is input to the plan constructor. This goal would be achieved by an instance of *accept-plan*, which decomposes into the surface speech action *s-accept*.

If the evaluation wasn't successful, then the goal of communicating the error is given to the plan constructor, where the error is simply represented by the node in the derivation that the evaluation failed at. There are two reasons why the evaluation could have failed, either no objects match, or more than one matches. In the first case, the referring expression is overconstrained, and the evaluation would have failed on an action that decomposes into surface speech actions. In the second case, the referring expression is underconstrained, and so the evaluation would have failed on the constraint that specifies the termination of the addition of modifiers. In our formalization

[7]We use the term *clarification*, since the conversational moves of judging and refashioning a referring expression can be viewed as clarifying it.

[8]See Levelt (1989, Chapter 12) for how prosody and clue words can be used in determining the type of clarification.

of the conversational moves, we have equated the first case to *reject-plan* and the second case to *postpone-plan*, and their constraints test for the abovementioned conditions. The actions *reject-plan* and *postpone-plan* decompose into the surface speech actions *s-reject* and *s-postpone*, respectively.

By observing the surface speech action corresponding to the judgment, the hearer, using plan inference, should be able to derive the speaker's judgment plan. If the judgment was *reject-plan* or *postpone-plan*, then the evaluation of the judgment plan should enable the hearer to determine the action in the referring plan that the speaker found problematic due to the constraints specified in the action schemas. The identify of the action in error will provide context for the subsequent refashioning of the referring expression.[9]

Refashioning Plans

If a conversant rejects a referring expression or postpones judgment on it, then either the speaker or the hearer will refashion the expression in the context of the rejection or postponement. In keeping with Clark and Wilkes-Gibbs, we use two discourse plans for refashioning: *replace-plan* and *expand-plan*. The first is used to replace some of the actions in the referring expression plan with new ones, and the second is to add new actions. Replacements can be used if the referring expression either overconstrains or underconstrains the choice of referent, while the expansion can be used only if it underconstrains the choice. So, these plans can check for these conditions.

The decomposition of the refashioning plans encode how a new referring expression can be constructed from the old one. This involves three tasks: first, a single candidate referent is chosen; second, the referring expression is refashioned; and third, this is communicated to the hearer by way of the action *s-actions*, which was already discussed.[10] The first step involves choosing a candidate. If the speaker of the refashioning is the agent who initiated the referring expression, then this choice is obviously pre-determined. Otherwise, the speaker must choose the candidate. Goodman (1985) has addressed this problem for the case of when the referring expression overconstrains the choice of referent. He uses heuristics to relax the constraints of the description and to pick one that *nearly* fits it. This problem is beyond the scope of this paper, and so we choose one of the referents arbitrarily (but see Heeman (1991) for how a simplified version of Goodman's algorithm that relaxes only a single constraint can be incorporated into the planning paradigm).

The second step is to refashion the referring expression so that it identifies the candidate chosen in the first step. This is done by using plan repair techniques (Hayes, 1975; Wilensky, 1981; Wilkens, 1985). Our technique is to remove the subplan rooted at the action in error and replan with another action schema inserted in its place. This technique has been encoded into our refashioning plans, and so can be used for both constructing repairs and inferring how another agent has repaired a plan.

Now we consider the effect of these refashioning plans. As we mentioned in section 2, once the refashioning plan is accepted, the common ground of the participants is updated with the new referring expression. So, the effect of the refashioning plans is that the hearer will believe that the speaker wants the new referring expression plan to replace the current one. Note that this effect does not make any claims about whether the new expression will in fact enable the successful identification of the referent. For if it did, and if the new referring expression were invalid, this would imply that the refashioning plan was also invalid, which is contrary to Clark and Wilkes-Gibbs's model of the acceptance process. So, the understanding of a refashioning does not depend on the understanding of the new proposed referring expression, but only on its derivation.

4.2 Action Schemas

This section presents action schemas for clarifications. Each clarification action includes a surface speech action in its decomposition. However, all reasoning is done at the level of the clarification

[9] Another approach would be to use the identity of the action in error to revise the beliefs that the agent has attributed to the other conversant and to use the revised beliefs in refashioning the plan. However, such reasoning is beyond the scope of this work.

[10] Another approach would have been to separate the communicative task from the first two (Lambert and Carberry, 1991).

actions, and so the surface actions do not include any constraints or effects. The notation used in the action schemas was given in Table 1 above.

accept-plan

The discourse action *accept-plan*, shown in Figure 8, is used by the speaker to establish the mutual belief that a plan will achieve its goal. The constraints of the schema specify that the plan being accepted achieves its goal and the decomposition is the surface speech action *s-accept*. The effect of the schema is that the hearer will believe that the speaker has the goal that it be mutually believed that the plan achieves its goal.

Header:	*accept-plan(Plan)*
Constraint:	*bel(Speaker,achieve(Plan,Goal))*
Decomposition:	*s-accept(Plan)*
Effect:	*bel(Hearer,goal(Speaker,bel(Hearer,bel(Speaker, achieve(Plan,Goal)))))*

Figure 8: *accept-plan* schema

reject-plan

The discourse action *reject-plan*, shown in Figure 9, is used by the speaker if the referring expression plan overconstrains the choice of referent. The speaker uses this schema in order to tell the hearer that the plan is invalid and which action instance the evaluation failed in. The constraints require that the error occurred in an action instance whose yield includes at least one primitive action. The decomposition consists of *s-reject*, which takes as its parameter the surface speech actions that are in the yield of the problematic action.

Header:	*reject-plan(Plan)*
Constraint:	*bel(Speaker,error(Plan,ErrorNode))*
	yield(Plan,ErrorNode,Acts)
	not(Acts = [])
Decomposition:	*s-reject(Plan,Acts)*
Effect:	*bel(Hearer,goal(Speaker,bel(Hearer,bel(System, error(Plan,ErrorNode)))))*

Figure 9: *reject-plan* schema

postpone-plan

The schema for *postpone-plan*, shown in Figure 10, is similar to *reject-plan*. However, it requires that the error in the evaluation occurred in an action that does not decompose into any primitive actions, which for referring expressions will be the instance of *modifiers* that terminates the addition of modifiers.

replace-plan

The *replace-plan* schema is used by the speaker to replace some of the primitive actions in a plan with new actions. Because we need knowledge of the type of action where the error occurred in order that we can refashion the invalid plan, the constraints of this schema are more specific than

Header:	*postpone-plan(Plan)*
Constraint:	*bel(Speaker,error(Plan,ErrorNode))*
	yield(Plan,ErrorNode,Acts)
	Acts = []
Decomposition:	*s-postpone(Plan,Acts)*
Effect:	*bel(Hearer,goal(Speaker,bel(Hearer,bel(Speaker, error(Plan,ErrorNode)))))*

Figure 10: *postpone-plan* schema

those of the judgment plans. The schema that we give in Figure 11, for instance, is used to refashion a referring expression plan in which the error occurred in an instance of a *modifier* action.[11]

Header:	*replace-plan(Plan)*
Constraint:	*bel(Speaker,error(Plan,ErrorNode))*
	content(Plan,ErrorNode,ErrorContent)
	ErrorContent = modifier(Entity,Object1,Cand,Cand1)
Decomposition:	*pick-one(Object,Cand)*
	Replacement = modifier(Entity,Object,Cand,Cand2)
	substitute(Plan,Node,Replacement,NewPlan)
	replan(NewPlan,Acts)
	s-actions(Plan,Acts)
Effect:	*bel(Hearer,goal(Speaker,bel(Hearer,bel(Speaker, replace(Plan,NewPlan)))))*

Figure 11: *replace-plan* schema

The decomposition of the schema specifies how a new referring expression plan can be built.[12] The first step, *pick-one(Object,Cand)*, chooses one of the objects that matched the part of the description that preceded the error; if the speaker is not the initiator of the referring expression, then this is an arbitrary choice. The second step specifies the header of the action schema that will be used to replace the subplan that contained the error. The third step substitutes the replacement into the referring expression plan, undoing all variable instantiations in the old plan. This results in the partial plan *NewPlan*. The fourth step calls the plan constructor to complete the partial plan. Finally, the fifth step is the surface speech action *s-actions*, which is used to inform the hearer of the surface speech actions that are being added to the referring expression plan.

expand-plan

The *expand-plan* schema, shown in Figure 12, is similar to the *replace-plan* schema shown in Figure 11. The difference is that instead of replacing one of the instances of *modifier*, it replaces the terminal instance of *modifiers* by a *modifiers* subplan that distinguishes one of the objects from the others that match, thus effecting an expansion of the surface speech actions. Even if the speaker thought that the referring expression as it stands were adequate (since the candidate set *Cand* contains only one member), she will construct a non-null expansion since the replacement is the recursive version of *modifiers*.

[11] If the error occurred in an instance of *headnoun*, a different *replace-plan* schema would need to be used, one that for instance relaxed the category that was used in describing the object (Goodman, 1985; Heeman, 1991).

[12] We refer to the steps in the decomposition that are not action headers as *mental actions*. They need to be proved, just like constraints.

Header:	*expand-plan(Plan)*
Constraint:	*bel(Speaker,error(Plan,ErrorNode))*
	content(Plan,ErrorNode,ErrorContent)
	ErrorContent = modifiers-terminate(Entity,Object1,Cand)
Decomposition:	*pick-one(Object,Cand)*
	Replacement = modifiers-recurse(Entity,Object,Cand)
	substitute(Plan,ErrorNode,Replacement,NewPlan)
	replan(NewPlan,Acts)
	s-actions(Plan,Acts)
Effect:	*bel(Hearer,goal(Speaker,mb(Speaker,Hearer,*
	replace(Plan,NewPlan))))

Figure 12: *expand-plan* schema

4.3 Plan Construction and Plan Inference

The general plan construction and plan inference processes are essentially the same as those for referring expressions. However, the plan inference process has been augmented so as to embody the criteria for understanding that were outlined in Section 4.1. The inference of judgment plans must be sensitive to the fact that such a plan includes the constraint that the speaker found the judged plan to be in error even though the hearer might not believe it to be. So, the inference process is allowed to assume that the speaker believes any constraint that the goal of the plan implies.

In the case of a refashioning, the hearer might not view the proposed referring expression plan as being sufficient for identifying the referent, but would nonetheless understand the refashioning. So, the inference process requires only that the proposed referring expression be derived—so that it can serve to replace the current plan—but not that it be acceptable. So, when a *replan* action is part of a plan that is being evaluated, the success of this action depends only on whether the plan that is its parameter can be derived, but not whether the derived plan is valid.[13]

5 Modeling Collaboration

In the last two sections, we discussed how initial referring expressions, judgments, and refashionings can be generated and understood in our plan-based model. In this section, we show how plan construction and plan inference fit into a complete model of how an agent collaborates in making a referring action successful. Previous natural language systems that use plans to account for the surface speech acts underlying an utterance (such as Cohen and Perrault, 1979; Allen and Perrault, 1980; Appelt, 1985; Litman and Allen, 1987) model only the recognition or only the construction of an agent's plans, and so do not address this issue.

In order to model an agent's participation in a dialogue, we need to model how the mental state of the agent changes as a result of the contributions that are made to the dialogue. The change in mental state can be modeled by the beliefs and goals that a participant adopts. When a speaker produces an utterance, as long as the hearer finds it coherent, he can add a belief that the speaker has made the utterance to accomplish some communicative goal. The hearer might then adopt some goal of his own in response to this, and make an utterance that he believes will achieve this goal. Participants expect each other to act in this way. These social norms allow participants to add to their common ground by adopting the inferences about an utterance as mutual beliefs.

To account for how conversants collaborate in dialogue, however, this co-operation is not strong enough. Not only must participants form mutual beliefs about what was said, they must also form mutual beliefs about the adequacy of the plan for the task they are collaborating upon. If

[13]Another approach would be to have the plan inference process reason about the intended effects of the plan that it is inferring in order to decide whether it should evaluate embedded plans and whether this evaluation should affect the evaluation of the parent plan.

the plan is not adequate, then they must work together to refashion it. This level of co-operation is due to what Clark and Wilkes-Gibbs refer to as a *mutual responsibility*, or what Searle (1990) refers to as a *we-intention*. This allows the agents to interact so that neither assumes control of the dialogue, thus allowing both to contribute to the best of their ability without being controlled or impeded by the other. This is different from what Grosz and Sidner (1990) have called master-servant dialogues, which occur in teacher-apprentice or information-seeking dialogues, in which one of the participants is controlling the conversation (cf. Walker and Whittaker, 1990). Note that the non-controlling agent may be helpful by anticipating obstacles in the plan (Allen and Perrault, 1980), but this is not the same as collaborating.

The mutual responsibility that the agents share not only concerns the goal they are trying to achieve, but also the plan that they are currently considering. This plan serves to coordinate their activity and so agents will have intentions to keep this plan in their common ground. The plan might not be valid (unlike the *shared plan* of Grosz and Sidner (1990)), so the agents might not mutually believe that each action contributes to the goal of the plan. Because of this, agents will have a belief regarding the validity of the plan, and an intention that this belief be mutually believed.

The discourse plans that we described in the previous section can now be seen as plans that can be used to further the collaborative activity. Judgment plans express beliefs about the success of the current plan, and refashioning plans update it. So, the mental state of an agent sanctions the adoption both of goals to express judgment and of goals to refashion. It also sanctions the adoption of beliefs about the current plan.[14] If it is mutually believed that one of the conversants believes there is an error with the current plan, the other also adopts this belief. Likewise, if one of the conversants proposes a replacement, the other accepts it. Since both conversants expect the other to behave in this way, each judgment and refashioning, so long as they are understood, results in the judgment or refashioning being mutually believed. Thus the current plan, through all of its refashionings, remains in the common ground of the participants.

Below, we discuss the rules for updating the mental state after a contribution is made. We then give rules that account for the collaborative process.[15]

5.1 Rules for Updating the Mental State

After a plan has been contributed to the conversation, by way of its surface speech actions, the speaker and hearer update their beliefs to reflect the contribution that has been made. Both assume that the hearer is observant, can derive a coherent plan (not necessarily valid), and can infer the communicative goal, which is expressed by the effect of the top-level action in the plan. We capture this by having the agent that we are modeling, the system, adopt the belief that it is mutually believed that the speaker intends to achieve the goal by means of the plan.[16]

bmb(system,user,plan(Speaker,Plan,Goal))

The system will also add a belief about whether she believes the plan will achieve the goal, and if not, the action that she believes to be in error. So, one of the following propositions will be adopted.

bel(system,achieve(Plan,Goal))
bel(system,error(Plan,Node))

After the above beliefs have been added, there are a number of inferences that the agents can make and, in fact, can believe will be made by the other participant as well, and so these inferences can be mutually believed. The first rule is that if it is mutually believed that the speaker intends to achieve *Goal* by means of *Plan*, then it will be mutually believed that the speaker has *Goal* as one

[14] The collaborative activity also sanctions discourse expectations that the other participant's utterances will pertain to the collaborative activity. We do not explicitly address this however.

[15] For simplicity, we represent the rules for entering into a collaborative activity, adopting beliefs, and adopting goals with the same operator, $\Longleftarrow$. For a more formal account, three different operators should be used.

[16] See Perrault (1990) for how these inferences can be drawn by using default rules.

of her goals.[17]

Rule 1
bmb(system,user,goal(Agt1,Goal)) $\Longleftarrow$
bmb(system,user,plan(Agt1,Plan,Goal)) &
Agt1 $\in$ {*system,user*}

The next rule concerns the adoption by the hearer of the intended goals of communicative acts. The communicative goal that we are concerned with is where the speaker wants the hearer to believe that the speaker believes some proposition. This only requires that the hearer believe the speaker to be sincere. We assume that both conversants are sincere, and so when such a communicative goal arises, both participants will assume that the hearer has adopted the goal. This is captured by rule (2).

Rule 2
bmb(system,user,bel(Agt1,Prop)) $\Longleftarrow$
bmb(system,user,goal(Agt1,bel(Agt2,bel(Agt1,Prop)))) &
Agt1,Agt2 $\in$ {*system,user*} *&*
not(Agt1 = Agt2)

The last rule involves an inference that is not shared. When the user makes a contribution to a conversation, the system assumes that the user believes that the plan will achieve its intended goal.

Rule 3
bel(system,bel(user,achieve(Plan,Goal))) $\Longleftarrow$
bmb(system,user,plan(user,Plan,Goal))

5.2 Rules for Updating the Collaborative State

The second set of rules that we give concern how the agents update the collaborative state. These rules have been revised from an earlier version (Heeman, 1991) so as to better model the acceptance process.

5.2.1 Entering into a Collaborative Activity

We need a rule that permits an agent to enter into a collaborative activity. We use the predicate *cstate* to represent that an agent is in such a state, and this predicate takes as its parameters the agents involved, the goal they are trying to achieve, and their current plan. Our view of when such a collaborative activity can be entered is very simple: the system believes it is mutually believed that one of them has a goal to refer and has a plan for doing so, but one of them believes this plan to be in error. The last part of the condition states that if the speaker's referring expression was successful from the beginning, no collaboration is necessary. It is not required that both participants mutually believe there is an error. Rather, if either detects an error, then that conversant can presuppose that they are collaborating, and make a judgment. Once the other recognizes the judgment that the plan is in error, the criteria for him entering will be fulfilled for him as well.

Rule 4
cstate(system,user,Plan,Goal) $\Longleftarrow$
bmb(system,user,goal(Agt1,Goal)) &
bmb(system,user,plan(Agt1,Plan,Goal)) &
Goal = knowref(Agt2,Agt1,Entity,Object) &
bel(system,bel(Agt3,error(Plan,Node))) &
Agt1,Agt2,Agt3 $\in$ {*system,user*} *&*
not(Agt1 = Agt2)

[17] All variables mentioned in the rules are existentially quantified.

5.2.2 Adoption of Mutual Beliefs

In order to model how the state of the collaborative activity progresses, we need to account for the mutual beliefs that the agents adopt as a result of the utterances that are made.

The first rule is for judgment moves in which the speaker finds the current plan in error. Given that the move is understood, both conversants, by way of the rules given in section 5.1, will believe that it is mutually believed that the speaker believes the current plan to be in error. In this case, the hearer, in the spirit of collaboration, must accept the judgment and so also adopt the belief that the plan is in error, even if he initially found the plan adequate. Since both conversants expect the hearer to behave in this way, the belief that there is an error can be mutually believed. Rule (5), below, captures this. (The adoption of this belief will cause the retraction of any beliefs that the plan is adequate.)

Rule 5
bmb(system,user,error(Plan,Node)) $\Longleftarrow$
cstate(system,user,Plan,Goal) &
bmb(system,user,bel(Agt1,error(Plan,Node))) &
Agt1 $\in$ {*system,user*}

The second rule is for refashioning moves. After such a move, the conversants will believe it mutually believed that the speaker has a replacement, *NewPlan*, for the current plan, *Plan*. Again, in the spirit of collaboration, the hearer must accept this replacement, and since both expect each other to behave this way, both adopt the belief that it is mutually believed that the new referring expression plan replaces the old one.

Rule 6
bmb(system,user,replace(Plan,NewPlan)) $\Longleftarrow$
Agt1 $\in$ {*system,user*} *&*
cstate(system,user,Plan,Goal) &
bmb(system,user,error(Plan,Node)) &
bmb(system,user,bel(Agt1,replace(Plan,NewPlan)))

In adopting this belief, the system updates the *cstate* by replacing the current plan with the new plan, and adding beliefs that capture the utterance of *NewPlan* as outlined in section 5.1 above.

The third rule is for judgment moves in which the speaker finds the current plan acceptable. Given that the move has been understood, each conversant will believe it is mutually believed that the speaker believes that the current plan will achieve the goal (second condition of the rule). However, in order to accept this move, each participant also needs to believe that the hearer also finds the plan acceptable (third condition). This belief would have been inferred if it were the hearer who had proposed the current plan, or the last refashioning. In this case, the speaker (of the acceptance) would have inferred by way of rule (3) that the hearer believes the plan to be valid; as for the hearer, given that he contributed the current plan, he undoubtedly also believes it to be acceptable.

Rule 7
bmb(system,user,achieve(Plan,Goal)) $\Longleftarrow$
cstate(system,user,Plan,Goal) &
bmb(system,user,bel(Agt1,achieve(Plan,Goal))) &
bel(system,bel(Agt2,achieve(Plan,Goal))) &
Agt1,Agt2 $\in$ {*system,user*} *&*
not(Agt1 = Agt2)

5.2.3 Adopting Goals

The last set of rules complete the circle. They account for how agents adopt goals to further the collaborative activity. These goals lead to judgment and refashioning moves, and so correspond to the rules that we just gave for adopting mutual beliefs.

The first goal adoption rule is for informing the hearer that there is an error in the current plan. The conditions specify that *Plan* is the current plan of a collaborative activity and that the speaker believes that there is an error in it.

Rule 8
goal(system,bel(user,bel(system,error(Plan,Node)))) $\Longleftarrow$
cstate(system,user,Plan,Goal) &
bel(system,error(Plan,Node))

The second rule is used to adopt the goal of replacing the current plan, *Plan*, if it has an error. The rule requires that the agent believe that it is mutually believed that there is an error in the current plan. So, this goal cannot be adopted before the goal of expressing judgment has been planned. Note that the consequent has an unbound variable, *NewPlan*. This variable will become bound when the system develops a plan to achieve this goal, by using the action schema *replace-plan* (Figure 11 above).

Rule 9
goal(system,bel(user,bel(system,replace(Plan,NewPlan)))) $\Longleftarrow$
cstate(system,user,Plan,Goal) &
bmb(system,user,error(Plan,Node))

The third rule is used to adopt the goal of communicating the system's acceptance of the current plan. Not only must the system believe that the plan achieves the goal, but it must also believe that the user also believes this. As mentioned above for rule (7), this last condition prevents the system from trying to accept a plan that it has itself just proposed. Rather, it can only try to accept a plan that the other agent contributed, for it is just such plans for which it will have the belief, by way of rule (3), that the user believes the plan achieves the goal.

Rule 10
goal(system,bel(user,bel(system,achieve(Plan,Goal)))) $\Longleftarrow$
cstate(system,user,Plan,Goal) &
bel(system,achieve(Plan,Goal)) &
bel(system,bel(user,achieve(Plan,Goal)))

5.3 Applying the Rules

The rules that we have given are used to update the mental state of the agent and to guide its activity. Acting as the hearer, the system performs plan inference on each set of actions that it observes, and then applies any rules that it can. When all of the observed actions are processed, the system switches from the role of hearer to speaker.

As the speaker, the system checks whether there is a goal that it can try to achieve, and if so, constructs a plan to achieve it. Next, presupposing its partner's acceptance of the plan, it applies any rules that it can. It repeats this until there are no more goals. The actions of the constructed plans form the response of the system; in a complete natural language system, they would be converted to a surface utterance. The system then switches to the role of hearer.

6 An Example

We are now ready to illustrate our system in action.[18] For this example, we use a simplified version of a subdialogue from the London-Lund corpus (Svartvik and Quirk, 1980, S.2.4a:1–8):

(6.1) **A:** [1] See the weird creature.
B: [2] In the corner?

[18]The system is implemented in C-Prolog under Unix.

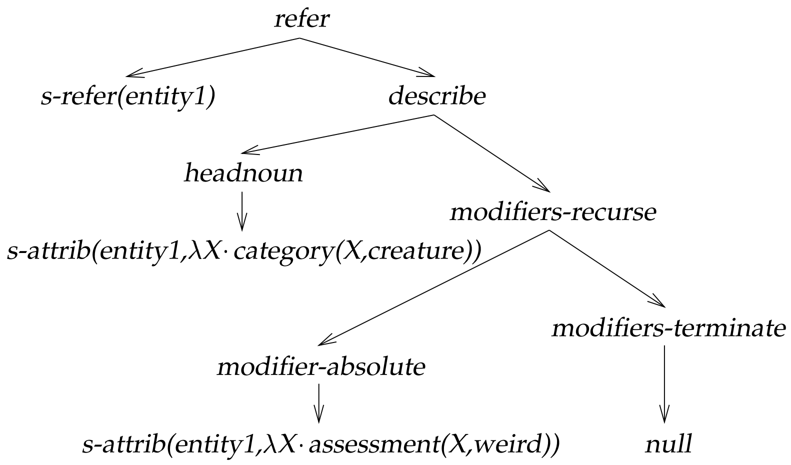

Figure 13: Plan derivation (*p1*) for "The weird creature"

A: [3] No, on the television.

B: [4] Okay.

The system will take the role of person B and we will give it the belief that there are two objects that are "weird"—a television antenna, which is on the television, and a fern plant, which is in the corner.

6.1 Understanding "The weird creature"

For the first sentence, the system is given as input the surface speech actions underlying "the weird creature," as shown below:

s-refer(entity1)
s-attrib(entity1,$\lambda X \cdot$ assessment(X,weird))
s-attrib(entity1,$\lambda X \cdot$ category(X,creature))

The system invokes the plan inference process, which finds the plan derivations whose yield is the above set of surface speech actions. In this case, there is only one, and the system labels it *p1*. Figure 13 shows the derivation; arrows represent decomposition, and for brevity, constraints and mental actions have been omitted and the parameters only of the surface speech actions are shown.

Next, the plan derivation is evaluated. The *subset* constraint in the *headnoun* action is evaluated, which narrows the candidate set to the antenna and the fern plant. The *subset* constraint in the *modifier* action is then evaluated, which does not eliminate either of the candidates, since the system finds both of them "weird." The constraint on the *modifiers* action that terminates the addition of modifiers is then evaluated. However, this constraint fails, since there are two objects that match the description rather than one, as required.

The system then updates its beliefs. As described in section 5.1, the system adds the following beliefs to capture the results of the plan inference process: that it is mutually believed that the user has the goal of *knowref* and has adopted *p1* as a means to achieve it, and that *p1* has an error on the terminating instance of *modifiers*, node *p22*.

bmb(system,user,plan(user,p1,knowref(system,user,entity1,Object))) (11)
bel(system,error(p1,p22)) (12)

The system next tries to apply the belief and goal adoption rules. From rule (1) and belief (11), the system adds the belief that it is mutually believed that the user has the goal that the system *knowref* and from rule (3) and belief (11), it adds the belief that the user believes that the plan achieves its goal.

bmb(system,user,goal(user,knowref(system,user,entity1,Object))) *(13)*
bel(system,bel(user,achieve(p1,knowref(system,user,entity1,Object)))) *(14)*

Belief (13), along with (11) and (12), allows the system to apply rule (4), and so the system enters into a collaborative activity, in which the goal is for it to know the referent and in which the current plan is *p1*.

cstate(system,user,p1,knowref(system,user,entity1,Object)) *(15)*

Since the system believes there is an error in the current plan, it applies rule (8), and so gives itself the communicative goal of informing the user of the error in the current plan.

goal(system,bel(user,bel(system,error(p1,p22)))) *(16)*

6.2 Constructing "In the corner?"

Since there are no further belief or goal adoption rules that can be applied, the system next checks for any goals that it can try to achieve. The only goal is (16), which is to inform the user of the error in the plan. Since the error in the referring plan is in the terminating instance of *modifiers*, the plan constructor builds an instance of *postpone-plan*, which it names *p26*. (The schema was given in Figure 10.) Rather than realizing the surface speech action immediately, the system plans ahead. (This would allow an opportunistic process to combine surface speech actions into a single utterance (Appelt, 1985).) So, the system, presupposing that the user understands the system's plan, adds the following belief.

bmb(user,system,plan(system,p26,bel(user,bel(system,error(p1,p22))))) *(17)*

It also adds the belief that this plan will achieve its goal.

bel(system,achieve(p26,bel(user,bel(system,error(p1,p22))))) *(18)*

Then by rule (1), the system adds the belief that it is mutually believed that it has the goal.

bmb(system,user,goal(system,bel(user,bel(system,error(p1,p22))))) *(19)*

Then by rule (2), which captures the co-operativity of the agents in communicative goals, it adds the belief that it is mutually believed that the system believes there is an error.

bmb(system,user,bel(system,error(p1,p22))) *(20)*

Then, on the basis of (15) and (20) the system applies rule (5), thus adopting the belief that it is mutually believed that there is an error in the plan. This presupposes the user's acceptance of the judgment plan.

bmb(system,user,error(p1,p22)) *(21)*

The system is now able to apply rule (9), on the basis of (15) and (21), and so adopts the goal of refashioning the invalid referring expression plan and of informing the user of the new plan.

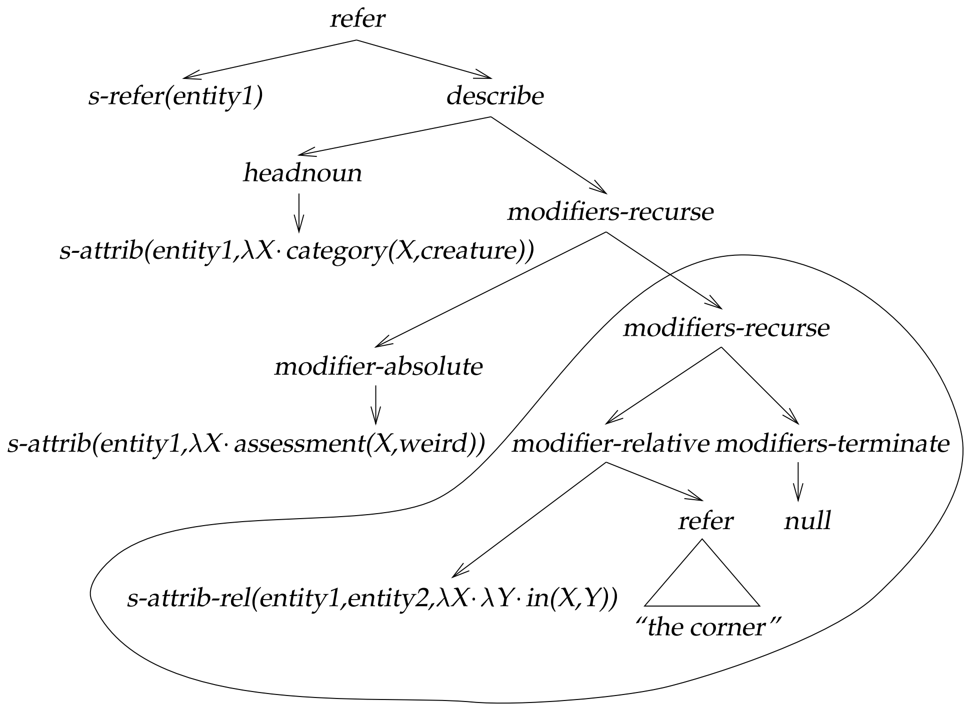

Figure 14: Plan derivation (*p34*) for "The weird creature in the corner"

goal(system,bel(user,bel(system,replace(p1,RPlan)))) *(22)*

Since no further rules can be applied, the system checks for goals that it can try to fulfill, which will result in choosing (22). To achieve this goal, the plan constructor builds an instance of *expand-plan* (previously shown in Figure 12). In doing this, the system chooses one of the objects that matched the original description as the likely referent; in this case it happens to choose the object in the corner, the fern plant, which the system represents as *fern1*. It then substitutes the *modifiers* subplan that terminates the addition of modifiers with the header of the *modifiers-recurse* action (with the chosen object instantiated in). The plan constructor is then called to fill in the details, thereby creating the expansion. The expansion it chooses includes a relative modifier (see Figure 7) that describes the object as being in the corner. The new referring plan (labeled *p34*) is shown in Figure 14, with the expansion circled (we have abbreviated the derivation of "the corner"). The surface speech action of *expand-plan* is *s-actions*, which takes the surface speech actions of the expansion, listed below, as its parameter.

s-attrib-rel(entity1,entity2,λX· λY· in(X,Y))
s-refer(entity2)
s-attrib(entity2,λX· category(X,corner))

Next, the system assumes the user will understand the refashioning, and, by way of rule (1) and (2), will be cooperative and adopt the communicative goal that the system believes that the new expanded plan replaces the old referring expression plan. The end result is given below as (23).

bmb(system,user,bel(system,replace(p1,p34))) *(23)*

The system, on the basis of (15) and (23), applies rule (6), and so assumes that the user will accept the refashioning. So, the system adds the belief that it is mutually believed that the new expanded plan replaces the old referring expression.

bmb(system,user,replace(p1,p34)) (24)

This causes the belief module to update the current plan of the collaborative activity (25). Also, it adds the beliefs that capture the utterance of the refashioned plan: that the system intends it as a means to achieve the referring action and that it does achieve this goal.[19]

cstate(system,user,p34,knowref(system,user,entity1,Object))) (25)
bmb(system,user,plan(system,p34,knowref(system,user,entity1,fern1))) (26)
bel(system,achieve(p34,knowref(system,user,entity1,fern1))) (27)

The two plans that were constructed, *postpone-plan* and *expand-plan*, give rise to the output of the surface speech actions *s-postpone* and *s-expand*, which would be realized as "in the corner?".[20]

6.3 Understanding "No, on the television"

The user next utters "No, on the television." This would get parsed into two separate surface speech actions, an *s-reject* corresponding to "no", and an *s-actions* corresponding to "on the television." For simplicity, the plan inference process is invoked separately on each.

The system starts with the *s-reject* action. We assume that the parser can determine from context that the "no" is rejecting the surface speech actions that were previously added and so the parameter of *s-reject* is a list of these actions. From this, it derives a plan whose yield is the *s-reject* action, and this plan is an instance of *reject-plan* (previously shown in Figure 9). The system then evaluates the constraints of the plan, which results in it determining which action in the plan the user found to be in error. This is done by evaluating the constraints of *reject-plan*, and so finding the action whose yield is the surface speech actions that were rejected. This will be *p56*, the *modifiers-relative* action that described the object as being in the corner. The resulting belief, after applying rules (1) and (2), is the following.

bmb(system,user,bel(user,error(p34,p56))) (28)

The system then applies the appropriate acceptance rule, rule (5), and so adopts the belief that the error is mutually believed.

bmb(system,user,error(p34,p56)) (29)

With this belief, the system will have the context that it needs to understand the user's refashioning plan.

The system next performs plan recognition starting with the second surface speech action, *s-actions*, which corresponds to the refashioning "on the television". It takes as a parameter the following list of actions:[21]

s-attrib-rel(entity1,entity3,$\lambda X\cdot\lambda Y\cdot$ *on(X,Y))*
s-refer(entity3)
s-attrib(entity3,$\lambda X\cdot$ *category(X,television))*

The system finds two plan derivations that account for the primitive action, one an instance of *replace-plan* (see Figure 11) and the other an instance of *expand-plan* (Figure 12). Next it evaluates the constraints of each derivation. The constraints of *expand-plan* do not hold since the action in error, *p56*, is not an instance of *modifiers-terminate*, so this plan is eliminated. The constraints

[19] Even though the system has the referent incorrectly identified in the goal of *knowref*, the goal itself is still valid: for it to identify the referent corresponding to *entity1*.
[20] Although our model does not account for the questioning intonation, it could be a manifestation of the *s-postpone*.
[21] We assume that the parser determines the appropriate discourse entities in these actions: *entity1* is the discourse entity for the object being referred to, and *entity3* is another discourse entity.

(and mental actions) of *replace-plan* do hold, and so the system is able to derive the refashioned referring plan, which it labels *p104*.

Since this instance of *replace-plan* is the only valid derivation corresponding to the surface speech actions observed, the system takes it as the plan behind the user's utterance. As a result, the system adds the following belief (after applying rule (1) and (2)).

bmb(system,user,bel(user,replace(p34,p104))) *(30)*

The system then applies the acceptance rule for refashioning plans, rule (6), and so adopts the refashioning as mutually believed.

bmb(system,user,replace(p34,p104)) *(31)*

This causes the belief module to update the current plan of the collaborative activity and to add the belief that the user contributed the new referring expression plan.

cstate(system,user,p104,knowref(system,user,entity1,Object)) *(32)*
bmb(system,user,plan(user,p104,knowref(system,user,entity1,antenna1)) *(33)*

The new referring plan will already have been evaluated. The subplan corresponding to "the television" would have been understood without problem,[22] and the modifier corresponding to "on the television" would have narrowed down the candidates that matched "weird creature" to a single object, *antenna1*. So, the belief module adds the belief that the system finds the new referring plan to be valid. Also, by way of rule (3), the system adds the belief that the user also does, since the user had proposed it.

bel(system,achieve(p104,knowref(system,user,entity1,antenna1))) *(34)*
bel(system,bel(user,achieve(p104,knowref(system,user,entity1,antenna1)))) *(35)*

6.4 Constructing "Okay"

On the basis of (32), (34), and (35), the system is able to apply rule (10), and so adopts the goal of accepting the plan.

goal(system,bel(user,bel(system,achieve(p104,
knowref(system,user,entity1,antenna1)))))) *(36)*

The plan constructor achieves this by planning an instance of *accept-plan*, which results in the surface speech action *s-accept*, which would be realized as "Okay." Then, after the application of rules (1), (2), and most importantly (7), the system adopts the belief that it is mutually believed that the plan achieves the goal of referring.

bmb(system,user,achieve(p104,knowref(system,user,entity1,antenna1))) *(37)*

7 Comparisons to Related Work

In providing a computational model of how agents collaborate upon referring expressions, we have touched on several different areas of research. First, our work has built on previous work in referring expressions, especially their incorporation into a model based on the planning paradigm.

[22]If "the television" is not understood, then since it is a referring expression in its own right, the conversants could collaborate on identifying its referent independently of the referent of "the weird creature"; that is, the participants could enter into an embedded collaborative activity by focusing on one part of the current plan.

Second, our work has built on the research done in modeling clarifications in the planning paradigm and on plan repair. Third, our work is related to the research being done on modeling collaborative and joint activity.

7.1 Referring Expressions

Cohen (1981) and Appelt (1985) have also addressed the generation of referring expressions in the planning paradigm. They have integrated this into a model of generating utterances, a step that we haven't taken. However, we have extended their model by incorporating even the generation of the components of the description into our planning model. One result of this is that our surface speech actions are much more fine-grained.

7.2 Clarifications and Plan Repair

An important part of our work involves accounting for clarifications of referring expressions by using meta-actions that incorporate plan repair techniques. This approach is based on Litman and Allen's work (1987) on understanding clarification subdialogues, in which meta-actions were used to model discourse relations, such as clarifications. There are several major differences between our work and theirs. First, our work addresses not only understanding but also generation and how these two tasks fit into a model of how agents collaborate in discourse. Second, Litman and Allen use a stack of unchanging plans to represent the state of the discourse. We, however, use a single *current plan*, modifying it as clarifications are made. This difference has an important ramification, for it results in different interpretations of the discourse structure. Consider dialogue (7.1), which was collected at an information booth in a Toronto train station (Horrigan, 1977). (Although the participants are not collaborating in making a referring expression, the dialogue will serve to illustrate our point.)

(7.1) **P:** [1] The 8:50 to Montreal?
C: [2] 8:50 to Montreal. Gate 7.
P: [3] Where is it?
C: [4] Down this way to your left. Second one on the left.
P: [5] OK. Thank you.

Litman and Allen represent the state of the discourse after the second utterance as a clarification of the passenger's *take-train-trip* plan. The information that the train boards at gate 7 is represented only in the clarification plan. So, when the passenger asks "Where is it?", their system, acting as the clerk, cannot interpret this as a clarification of the *take-train-trip* plan, since the utterance "cannot be seen as a step of [that] plan" (p. 188). So, it is interpreted instead as a request for a clarification of the clerk's "Gate 7" response, implicitly assuming that "Gate 7" was not accepted. In our model, the acceptance of "Gate 7" would be presupposed, and so it would be incorporated into the *take-train-trip* plan. So, the passenger's question of "Where is it?" would be viewed as a request for the clerk to clarify that plan.

The work of Moore and Swartout (1991), Cawsey (1991), and Carletta (1991) on interactive explanations also addresses clarifications using plan repair techniques. This body of work uses plan construction techniques to generate explanations, and uses the constructed plan as a basis for recovery strategies if the user doesn't understand the explanation. In the cases of Cawsey and Carletta, both use meta-actions to encode the plan repair techniques. However, none of these approaches are within a collaborative framework, in which either agent can contribute to the development of the plan.

Other relevant work is that of Lambert and Carberry (1991). In their model of understanding information-seeking dialogues, they propose a distinction between problem-solving activities and discourse activities. In contrast, our clarifications embody both functions in the same actions, thus allowing for a simpler approach to inferring the refashioned referring expressions, since we need not chain to a meta-operator. In later work, Chu-Carroll and Carberry (1994) extended this model to generate responses to proposals that are viewed as sub-optimal or invalid. Like Litman and Allen (1987), they adopt the view that subsequent modifications apply to the preceding modification, rather than the underlying plan.

7.3 Collaboration

Grosz, Sidner, and Lochbaum (Grosz and Sidner, 1990; Lochbaum, Grosz, and Sidner, 1990) are interested in the type of plans that underlie discourse in which the agents are collaborating in order to achieve some goal. They propose that agents are building a *shared plan* in which participants have a collection of beliefs and intentions about the actions in the plan. Our model differs from theirs in two important aspects. First, not only do agents have a collection of beliefs and intentions regarding the actions of a shared plan, we feel that they also have an intention about the goal (Searle, 1990; Cohen and Levesque, 1991). It is this intention, in conjunction with the current plan, that sanctions the adoption of beliefs and intentions about potential actions that will contribute to the goal, rather than just the shared plan.

Second, we feel that their definition of a partial shared plan is too restrictive. Although they address partial plans, they require, in order for an action to be part of a partial shared plan, that both agents believe that the action *contributes* to the goal. However, this is too strong. In collaborating to achieve a mutual goal, participants sometimes propose an action that is not believed by the other participant or even by the participant that is proposing it. In failing to represent such states, their model is unable to represent the intermediate states in which a hearer might have understood how the speaker's utterance contributes to a plan, but doesn't agree with it. This is important, since if the refashioned plan is invalid, only the referring expression should be refashioned, not the refashioning itself.

Traum (1991; Traum and Hinkelman, 1992) is concerned with providing a computational model of *grounding*, the process in which conversational participants add to the common ground of a conversation (Clark and Schaefer, 1989; Clark and Brennan, 1990). Traum models the grounding process by proposing that utterances move through a number of states, 'pushed' by grounding acts, which include initiate, continue, repair, request repair, acknowledge, and request acknowledge. Once an utterance has been acknowledged, it will reside in mutual belief as a proposal of the person who initiated it. The proposal state is a subspace of the mutual belief space of the conversants. Only once it has been accepted, will it be moved into the *shared* space (also in mutual belief). Unlike Traum's, our work does not differentiate the proposal state from the shared state. If a proposal is understood, it is incorporated into the current plan. Judgments of acceptability are not on proposals but on the current plan, or a part of it.

Sidner (1992) addressed the issue of how conversational participants collaborate in building a shared plan. In this work, Sidner presents a number of speech actions for use in collaborative tasks. These actions are those that an artificial agent could use in negotiating which actions or beliefs to accept into the shared plan of the agents. As with Traum, it is the *proposals* that are refashioned, before they are integrated into the shared plan, rather than the shared plan.

Cohen and Levesque (1991) focus on formalizing joint intention in a logic. They use this formalism to explain how such elements of communication as confirmations arise when agents are engaging in a joint action. However, they have not addressed how agents collaborate in building a plan, only how agents collaborate while executing a plan. Once this limitation is overcome, their approach could offer us a route for formalizing the mental states of the collaborating agents in our model and for proving that our acceptance and goal adoption rules follow from such states.

8 Conclusion

We have presented a computational model of how a conversational participant collaborates in making and understanding a referring expression, based on the view that language is goal-oriented behavior. This has allowed us to do the following. First, we have accounted for the tasks of building a referring expression and identifying its referent by using plan construction and plan inference. Second, we have accounted for the conversational moves that participants make during the acceptance process by using meta-actions. Third, we have accounted for collaborative activity by proposing that agents are in a certain mental state that includes a goal, a plan that they are currently considering, and intentions. This mental state sanctions the acceptance of clarification plans, and sanctions the adoption of goals to clarify. Although our work has focused on referring expressions, we feel that it is relevant to collaboration in general and to how agents contribute to discourse.

This paper is based on the model of collaboration proposed by Clark and Wilkes-Gibbs (1986). Their model makes two strong claims about how agents collaborate. First, it minimizes the distinction between the roles of the person who initiates the referring expression and the person who is trying to identify it. Both have the same moves available to them, for either can judge the description and either can refashion it. This allows both participants to contribute without being controlled or impeded by the other. Second, their model gives special status to the role of the current referring expression (current plan): participants judge and refashion the current referring expression directly, rather than recursively modifying modifications (e.g. Litman and Allen, 1987; Chu-Carrol and Carberry, 1994) or incrementally adding to the current plan with each accepted proposal (e.g. Traum and Hinkelman, 1992; Sidner, 1992). In our work, we have taken Clark and Wilkes-Gibbs's descriptive model and recast it into a computational one, thus demonstrating the computational feasibility of their work and its compatibility with current practices in artificial intelligence.

There are many ways that this research could be extended. Perhaps the most obvious would be to extend the planning component of our model. First, our coverage of referring expressions could be extended to handle references to objects in focus and to descriptions that include a plan of physical actions for identifying the referent. Second, the treatment of clarifications could be improved; specifically, how plan failures are reasoned about, how plan failures affect the agent's beliefs, and how these failures are repaired. Third, this research needs to be integrated into a more complete plan-based approach to language, and needs to be extended so as to handle more general discourse plan failures (McRoy and Hirst, 1993; McRoy and Hirst, 1994; Horton and Hirst, 1991; Heeman, 1993; Edmonds, 1994; Hirst et al., 1994). A benchmark for such future work could be dialogue (8.1) below, from the London-Lund corpus (Svartvik and Quirk, 1980, S.2.4a:1–8), which is the basis of the example used in section 6. This dialogue shows how collaboration on a referring expression can be embedded in other activities, how agents can return back to a collaborative activity, and even how agents can take advantage of a mistaken referent.

(8.1) **A:** [1] What's that weird creature over there?

B: [2] In the corner?

A: [3] *affirmative noise*

B: [4] It's just a fern plant.

A: [5] No, the one to the left of it.

B: [6] That's the television aerial. It pulls out.

A second avenue for future work is to further investigate collaborative behavior and protocols for interaction. We need to formalize what it means for agents to be collaborating, in a theory that takes account of rational interaction and the beliefs and knowledge of the participants. Such a theory would do the following. First, it would give a more complete motivation for the processing rules that we used for how agents interact in a collaborative activity. Second, it would account for why agents would enter into such a mode of interaction, how it is initiated, how it is carried forward (especially how agents' beliefs and knowledge influence their actions), and how it ends. Third, it would be extendable to other forms of interaction, such as information-seeking dialogues. Fourth, it would specify how collaborative activity could be embedded in, or embed, other types of interactions. By answering these questions, we will not only have a better model to base natural language interfaces on, but we will also have a better understanding of how people interact.

Acknowledgments

We would like to thank James Allen, Hector Levesque, and the referees at *Computational Linguistics* for their comments on an earlier version of this paper. We would also like to especially thank Janyce Wiebe for her invaluable contribution to the development of this work. As well, we are grateful for comments from, and discussions with, Diane Horton, Susan McRoy, Massimo Poesio, and David Traum. Funding at the University of Toronto and the University of Rochester was provided by the Natural Sciences and Engineering Research Council of Canada, with additional funding at Rochester provided by NSF under Grant IRI-90-13160 and ONR/DARPA under Grant N00014-92-J-1512.

References

Allen, James, James Hendler, and Austin Tate, editors. 1990. *Readings in Planning*. Morgan Kaufmann Publishers.

Allen, James F. and C. Raymond Perrault. 1980. Analyzing intention in utterances. *Artificial Intelligence*, 15:143–178. Reprinted in (Grosz, Sparck Jones, and Webber, 1986).

Appelt, Douglas and Amichai Kronfeld. 1987. A computational model of referring. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI '87)*, pages 640–647.

Appelt, Douglas E. 1985. Planning English referring expressions. *Artificial Intelligence*, 26(1):1–33, April. Reprinted in (Grosz, Sparck Jones, and Webber, 1986).

Austin, J. L. 1962. *How to do things with words*. New York: Oxford University Press.

Carletta, Jean. 1991. Recovering from plan failure using a layered architecture. Research Paper 524, Department of Artificial Intelligence, University of Edinburgh.

Cawsey, Alison. 1991. Generating interactive explanations. In *Proceedings of the National Conference on Artificial Intelligence (AAAI '91)*, pages 86–91.

Chu-Carroll, Jennifer and Sandra Carberry. 1994. A plan-based model for response generation in collaborative task-oriented dialogues. In *Proceedings of the National Conference on Artificial Intelligence (AAAI '94)*.

Clark, Herbert H., editor. 1992. *Arenas of Language Use*. University of Chicago Press and CSLI.

Clark, Herbert H. and S. E. Brennan. 1990. Grounding in communication. In L.B. Resnick, J. Levine, and S.D. Behreno, editors, *Perspectives on Socially Shared Cognition*. APA.

Clark, Herbert H. and Catherine R. Marshall. 1981. Definite reference and mutual knowledge. In Aravind K. Joshi, Bonnie Lynn Webber, and Ivan Sag, editors, *Elements of Discourse Understanding*. Cambridge University Press, Cambridge, pages 10–62.

Clark, Herbert H. and Edward F. Schaefer. 1989. Contributing to discourse. *Cognitive Science*, 13:259–294. Reprinted in (Clark, 1992), pages 144–175.

Clark, Herbert H. and Deanna Wilkes-Gibbs. 1986. Referring as a collaborative process. *Cognition*, 22:1–39. Reprinted in (Clark, 1992), pages 107–143.

Cohen, Philip R. 1981. The need for referent identification as a planned action. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI '81)*, pages 31–36.

Cohen, Philip R. and Hector J. Levesque. 1991. Confirmation and joint action. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI '91)*.

Cohen, Philip R. and C. Raymond Perrault. 1979. Elements of a plan-based theory of speech acts. *Cognitive Science*, 3(3):177–212. Reprinted in (Grosz, Sparck Jones, and Webber, 1986).

Dale, R. 1989. Cooking up referring expressions. In *Proceedings of the 27th Annual Meeting of the Association for Computational Linguistics*, pages 68–75.

Edmonds, Philip G. 1994. Collaboration on reference to objects that are not mutually known. In *Proceedings of the 15th International Conference on Computational Linguistics (COLING '94)*, Kyoto.

Goodman, Bradley A. 1985. Repairing reference identification failures by relaxation. In *Proceedings of the 23rd Annual Meeting of the Association for Computational Linguistics*, pages 204–217.

Grosz, Barbara J. and Candace L. Sidner. 1990. Plans for discourse. In Philip R. Cohen, Jerry Morgan, and Martha E. Pollack, editors, *Intentions in Communication*, SDF Benchmark Series. MIT Press, pages 417–444.

Grosz, Barbara J., Karen Sparck Jones, and Bonnie Lynn Webber, editors. 1986. *Readings in Natural Language Processing*. Morgan Kaufmann Publishers.

Hayes, Philip J. 1975. A representation for robot plans. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI '75)*, pages 181–188. Reprinted in (Allen, Hendler, and Tate, 1990).

Heeman, Peter A. 1993. Speech actions and mental states in task-oriented dialogues. In *Working Notes AAAI Spring Symposium on Reasoning about Mental States: Formal Theories & Applications*, pages 68–73, Stanford, March.

Heeman, Peter Anthony. 1991. A computational model of collaboration on referring expressions. Master's Thesis, Technical Report CSRI 251, Department of Computer Science, University of Toronto, September.

Hirst, Graeme, Susan McRoy, Peter Heeman, Philip Edmonds, and Diane Horton. 1994. Repairing conversational misunderstandings and non-understandings. *Speech Communications*. To Appear.

Horrigan, Mary Katherine. 1977. Modelling simple dialogs. Master's Thesis, Technical Report 108, Department of Computer Science, University of Toronto, May.

Horton, Diane and Graeme Hirst. 1991. Discrepancies in discourse models and miscommunication in conversation. In *Working Notes of the AAAI symposium: Discourse Structure in Natural Language Understanding and Generation*, pages 31–32.

Kautz, Henry A. and James F. Allen. 1986. Generalized plan recognition. In *Proceedings of the National Conference on Artificial Intelligence (AAAI '86)*, pages 32–37.

Lambert, Lynn and Sandra Carberry. 1991. A tripartite plan-based model for dialogue. In *Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics*, pages 47–54.

Levelt, Willem J. M. 1989. *Speaking: from intention to articulation*. Cambridge: Cambridge University Press.

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

Lochbaum, Karen E., Barbara J. Grosz, and Candace L. Sidner. 1990. Models of plans to support communication: An initial report. In *Proceedings of the National Conference on Artificial Intelligence (AAAI '90)*, pages 485–490.

McRoy, Susan and Graeme Hirst. 1993. Abductive explanations of dialogue misunderstanding. In *Proceedings, 6th Conference of the European Chapter of the Association for Computational Linguistics*, pages 277–286, Utrecht, April.

McRoy, Susan and Graeme Hirst. 1994. The repair of speech act misunderstandings by abductive inference. Submitted for publication.

Mellish, C. S. 1985. *Computer Interpretation of Natural Language Descriptions*. Ellis Horwood Series in Artificial Intelligence. Chichester, West Sussex, England: Ellis Horwood.

Moore, Jahanna D. and William R. Swartout. 1991. A reactive approach to explanation: taking the user's feedback into account. In Cécile L. Paris, William R. Swartout, and William C. Mann, editors, *Natural Language Generation in Artificial Intelligence and Computational Linguistics*. Kluwer Academic Publishers, pages 3–48.

Nadathur, Gopalan and Aravind K. Joshi. 1983. Mutual beliefs in conversational systems: Their role in referring expressions. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI '83)*, pages 603–605.

Perrault, C. R. 1990. An application of default logic to speech act theory. In Philip R. Cohen, Jerry Morgan, and Martha E. Pollack, editors, *Intentions in Communication*, SDF Benchmark Series. MIT Press, pages 161–185.

Perrault, C. Raymond and Philip R. Cohen. 1981. It's for your own good: a note on inaccurate reference. In Aravind K. Joshi, Bonnie Lynn Webber, and Ivan Sag, editors, *Elements of Discourse Understanding*. Cambridge University Press, Cambridge, pages 217–230.

Pollack, Martha E. 1990. Plans as complex mental attitudes. In Philip R. Cohen, Jerry Morgan, and Martha E. Pollack, editors, *Intentions in Communication*, SDF Benchmark Series. MIT Press, pages 77–103.

Reiter, Ehud. 1990. The computational complexity of avoiding conversational implicature. In *Proceedings of the 28th Annual Meeting of the Association for Computational Linguistics*, pages 97–104.

Searle, J. R. 1969. *Speech acts: An essay in the philosophy of language*. Cambridge: Cambridge University Press.

Searle, John R. 1990. Collective intentions and actions. In Philip R. Cohen, Jerry Morgan, and Martha E. Pollack, editors, *Intentions in Communication*, SDF Benchmark Series. MIT Press, pages 401–415.

Sidner, Candace L. 1985. Plan parsing for intended response recognition in discourse. *Computational Intelligence*, 1(1):1–10.

Sidner, Candace L. 1992. Using discourse to negotiate in collaborative activity: An artificial language. In *Proceedings of the Workshop on Cooperation among Heterogeneous Intelligent Agents*. AAAI-'92.

Svartvik, J. and R. Quirk. 1980. *A Corpus of English Conversation*. Lund Studies in English. 56. Lund: C.W.K. Gleerup.

Traum, David R. 1991. Towards a computational theory of grounding in natural language conversation. Technical Report 401, Department of Computer Science, University of Rochester.

Traum, David R. and Elizabeth A. Hinkelman. 1992. Conversation acts in task-oriented spoken dialogue. *Computational Intelligence*, 8(3).

Vilain, Marc. 1990. Getting serious about parsing plans: a grammatical analysis of plan recognition. In *Proceedings of the National Conference on Artificial Intelligence (AAAI '90)*, pages 190–197.

Walker, Marilyn and Steve Whittaker. 1990. Mixed initiative in dialogue: An investigation into discourse segmentation. In *Proceedings of the 28th Annual Meeting of the Association for Computational Linguistics*, pages 70–78.

Webber, Bonnie Lynn. 1983. So what can we talk about now? In Michael Brady and Robert C. Berwick, editors, *Computational Models of Discourse*. MIT Press, Cambridge, pages 331–371. Reprinted in (Grosz, Sparck Jones, and Webber, 1986).

Wilensky, Robert. 1981. A model for planning in complex situations. *Cognition and Brain Theory*, 4. Reprinted in (Allen, Hendler, and Tate, 1990).

Wilkens, David E. 1985. Recovering from execution errors in SIPE. *Computational Intelligence*, 1:33–45. Reprinted in (Allen, Hendler, and Tate, 1990).