Representing Communicative Intentions in Collaborative Conversational Agents

Matthew Stone
Department of Computer Science and Center for Cognitive Science
Rutgers, the State University of New Jersey
110 Frelinghuysen Road, Piscataway NJ 08854-8019
mdstone@cs.rutgers.edu

Abstract

This paper pursues a formal analogy between natural language dialogue and collaborative real-world action in general. The analogy depends on an analysis of two aspects of collaboration that figure crucially in language use. First, agents must be able to coordinate *abstractly* about future decisions which cannot be made on present information. Second, when agents finally take such decisions, they must again coordinate in order to *interpret* one anothers' actions as collaborative. The contribution of this paper is a general representation of collaborative plans and intentions, inspired by representations of deductions in logics of knowledge, action and time, which supports these two kinds of coordination. Such representations can describe natural language dialogue simply by specifying the potential that utterances have, in virtue of their meanings, to contribute to an evolving record of the conversation. These representations are implemented in a simple prototype collaborative dialogue agent.

Introduction

When people talk to one another face-to-face to accomplish real-world tasks, they recruit a range of behaviors and make extensive use of the real-world environment around them, in a joint effort to maintain a shared understanding of one anothers' contributions to the dialogue and to the ongoing task. In human-human dialogue, this situated and collaborative effort is manifest in phenomena such as the following:

- *Disambiguation based on plan-recognition.* People draw on knowledge of their partner's perspective, goals and beliefs to constrain interpretation problems such as reference resolution on-line (Hanna and Tanenhaus 2001).
- *Clarification subdialogues and negotiation of meaning.* People continue communicating until they share an understanding of each utterance that is sufficient for the purposes at hand (Clark and Wilkes-Gibbs 1986; Clark and Schaefer 1989; Brennan 1990).
- *Accommodation and cooperative response.* When people recognize that their partner's utterance depends on unexpected assumptions, they can adjust the model of the conversation implicitly by accommodating, or taking on, those new assumptions (Lewis 1979; Thomason 1990), and they can raise discussion about those assumptions and offer corrections to them (Cheikes 1991).
- *Entrainment.* Participants in conversations come to agree on the vocabulary they use to communicate (Brennan and Clark 1996; Brennan 1996).
- *Multimodality.* People's speech co-occurs with other communicative behaviors, such as gesture and facial displays, which contribute to a single consistent utterance interpretation (Cassell *et al.* 1999; Cassell 2000).

Such phenomena provide compelling evidence for basing a cognitive science of language use on a systematic analogy between dialogue and collaborative agency (Clark 1996). However, many challenges remain in formalizing language use and collaborative real-world action in parallel ways, and in implementing such a formalism for collaborative conversational agents. Take the simple question-and-answer exchange in (1), for example.

(1) a X: What size do you want: small or large?
   b Y: Large.

(1) exhibits entrainment; in (1b) Y answers the question in (1a) in the terms in which it is posed. The response shows the extent of collaboration in dialogue, in that Y acts to meet not only X's expectation for the content of the response (that Y will identify a desired size), but also X's expectation for the form of the response (identifying one size as *small* and the other as *large*).

This view of (1) is representative of collaborative accounts of language use in depending on two abilities that agents bring to collaboration. First, agents can coordinate *abstractly* about future decisions which cannot be made solely on currently shared information. In (1), for example, X and Y coordinate on the answer Y is to give—something that X cannot know exactly. Second, agents can coordinate to *interpret* one anothers' actions as collaborative. In (1), X and Y can understand each utterance as evoking specific discourse referents, or as moving the conversation forward, only by bringing to bear expectations derived from their shared understanding of their environment and their ongoing dialogue.

In this paper, I describe a general representation of collaborative plans and intentions that supports these two kinds of coordination. These representations abstract safely over future actions by incorporating features of deductions in logics of knowledge, action and time (Moore 1985; Stone 1998) so as to anticipate and describe the choices agents can make in carrying out an intention. At the same time, these representations can serve as resources for deliberation, coordination and action, in the spirit of philosophers such as Bratman (Bratman 1987) and computer scientists such as Pollack (Pollack 1992). In particular, it is a straightforward matter for agents to follow through on their collaborative intentions, or to recognize how their teammates are doing so.

Such representations can describe natural language dialogue simply by specifying the potential that utterances have, in virtue of their meanings, to contribute to an evolving record of the conversation. Doing so achieves a broadly Gricean formalization of language use as intentional activity (Grice 1957), yet at the same time it retains important insights from formal semantics about the dynamics and context-dependence of meaning (van der Sandt 1992; Kamp and Reyle 1993; Muskens 1996).

The model is realized in a simple prototype dialogue agent, and enables the prototype to exhibit a variety of situated and collaborative features of conversation, including plan-recognition in interpretation, entrainment in generation, and cooperative response in dialogue management.

A Framework for Agency and Intention

For the purposes of this paper, I understand an *intention* as a *plan* that an agent is *committed to*. Here, a plan represents an *argument* that demonstrates that an agent can perform some series of actions to obtain a good outcome. Committing to it means undertaking, within certain limits, to perform the plan's actions and so realize the plan's outcome, for the very reasons the plan spells out.

This framework balances the two roles of intention in collaboration. On the one hand, agents must be able to draw on their own intentions to facilitate their deliberation and action in a complex, uncertain and changing world (Bratman 1987; Pollack 1992). On the other hand, one agent's attribution of an intention to another agent, like our own folk attributions of intention (Malle and Knobe 1997), must encapsulate a rich understanding of that agent's goals, beliefs and deliberation (Pollack 1990).

Consider first deliberation. Let us adopt the standard idealization of an agent's cognition as a series of cycles; in each cycle, the agent first perceives its environment, then deliberates to update its desires, beliefs and intentions, and finally carries out an action (Bratman *et al.* 1988). Within this architecture, any intention that an agent carries into a new cycle of perception and action provides the agent with an argument about what to do. The argument sets out expectations, governed by a set of defeasible assumptions, for the way the agent may perceive the world to be. When these expectations are met, the argument specifies a reason to act, consisting of an action that the agent can perform now, and perhaps some further intentions that can guide the agent into the future cycles of perception and action. And even when those expectations fail, the intention can serve as a reference point for repairing the agent's strategy. Intentions thus serve as a resource for deliberation, by focusing the agent's reasoning on a constellation of beliefs, actions and desires that governs the agent's immediate success in its environment.

Attributions of intention, meanwhile, exploit the same constellation of attitudes. Suppose agent X performs an action A, and agent Y recognizes the intention behind X's doing A. That means, of course, that Y attributes X's action to X's commitment to an argument about the effects of doing A. The argument involves assumptions about the state of the world and the causal consequences of A—assumptions that X must endorse. The argument also describes an outcome—one that X must favor. Without these attitudes, X would not have made and persisted in a commitment to this intention. Intention thus brings with it a complex ensemble of beliefs and desires describing not only a desired action but more generally that action's circumstances and effects.

I am inclined to emphasize the parallels between this framework and other models of agency: the notion of plans as arguments from (Ferguson and Allen 1994); the notions of choice and commitment in intention from (Cohen and Levesque 1990); Pollack's understanding of intention as a complex mental attitude (Pollack 1990); and the belief-desire-intention architectures for rationality explored generally in such work as (Bratman *et al.* 1988; Rao and Georgeff 1992). But only up to a point. For in the present framework, unlike in others, the paramount question is *what kinds of formal structures can represent intentions as arguments*. To pursue this question, we must turn to representations of inference in computational logic.

Representing Intention as Inference

Ordinarily, we think of inference as a process, as an activity which creates new true statements by rewriting old ones, perhaps. However, because the process of inference proceeds in discrete steps governed by formal rules, we can also collect together the statements involved in an inference as a single structured syntactic object. We might encode such an object as a *proof-term* in a formal specification language, or in another such computational data structure. In so doing, we obtain a representation of the inference as a whole.

To record many kinds of argumentation, proof-terms must be constructed from *typed* elements using *variable-binding* operations. Variables and binding describe inferences which take on temporary assumptions for the purposes of argument. Type-disciplines ensure that such constructions are permitted only when licensed by the logic.

Perhaps the clearest illustration of these ideas comes in the Curry-Howard isomorphism, which shows that natural deduction proofs in Intuitionistic logic can be represented using terms in the simply-typed λ-calculus (Gallier 1993). Intuitionistic logic strengthens implication to give it a constructive force with a natural relevance to computation and decision-making. In an Intuitionistic proof of $P \supset Q$, we can use P as an assumption only locally, as part of this proof of Q. Accordingly, a proof-term for such an inference combines an indication of the rule used (*implication introduction*, or *ii*) and a λ-term representing a function mapping any proof for P into a proof for Q. For example, $P \supset P$ is proved by $ii(\lambda p.p)$. Here $\lambda p.p$ is the identity function that simply delivers the assumed proof of P as the required proof of P. Types and variable-binding are by no means limited to Intuitionistic natural deduction, of course. For example, (Pfenning 2000) relies on them to develop simple proof-terms for sequent calculus proofs, and for classical logic.

To take advantage of such representations, I will be using the higher-order logic programming language λProlog (Nadathur and Miller 1998). λProlog offers functions as data structures using the simply-typed λ-calculus and includes higher-order unification to reason about these data structures in a principled way. In the syntax of λProlog, function expressions $\lambda x.M$ are encoded as `x\ M`, and these λ-terms see general use as the arguments of logical operators that bind variables (such as *ii* above). The use of logic programming makes it possible to specify agents declaratively, and to reuse these specifications for multiple tasks. For dialogue, for example, the same intention representations and linguistic knowledge can be applied both for understanding and for generation. (See (Pereira and Shieber 1987).) Meanwhile, although general higher-order unification is explosive, (Miller 1991) shows that this expressive power need not introduce much complexity over standard Prolog, when functions and bound variables are used in constrained ways.

In the remainder of this section, I first summarize the account of planning as an inference problem from (Stone 1998). I then introduce concrete syntax to specify such plan inferences as λProlog terms. Finally, I show how these data structures can guide an agent's deliberation and action.

Formalizing Knowledge, Action and Choice

Temporal and modal logics provide a natural setting for specifying a changing world and an agent's changing information in the world (Halpern and Moses 1985; Fagin *et al.* 1995; Fitting and Mendelsohn 1998). These languages extend classical logic by adding operators that characterize propositions that are known to an agent or that are true at some point in time. Here, I use the two kinds of operators in (2):

(2) a $[\mathrm{X}]P$
b x DOES a THEN P

$[\mathrm{X}]P$ means that agent X knows P. (I'll also use λProlog syntax `k x P`.) x DOES a THEN P means that if agent X does action A now, P will be true immediately afterwards.

To characterize an agent's deliberation across several cycles of perception and action, we must be able to describe the new information that will become available in future situations. Such descriptions depend on existential quantifiers and disjunctions (Hintikka 1971). Thus, consider (3).

(3) a $[\mathrm{X}]\exists a.Pa$
b $\exists a.[\mathrm{X}]Pa$

(3a) gives a specific characterization of agent X's indefinite knowledge: X knows that something has property P, but not necessarily what. Meanwhile, (3b) gives an indefinite characterization of agent X's specific knowledge; there is some real-world object A which X knows to have property P, but the formula does not say what A is. The formulas of (3) can be encoded as in (4) using λ-terms to abstract bound variables, and using `sm` for the existential quantifier.

(4) a `k x (sm a\[p a])`
b `sm a\[k x (p a)]`

In planning, sensing actions can move an agent from an epistemic state characterized as in (3a) to a state characterized as in (3b). Representing the difference makes it possible to anticipate an agent's future choice of an appropriate action. For example, consider an intention that an agent X carries from one cycle of deliberation to the next, which maps out X's choice of a single future action A. In the next stage, X will choose A based on the information that X has then. X's information should indicate that A will clearly bring about a desired state G. This condition is formalized in (5).

(5) $\exists a[\mathrm{X}](x \text{ DOES } a \text{ THEN } [\mathrm{X}]G)$

If X knows now that (5) will hold, X knows that the next step of deliberation will be successful. However, with only an indefinite characterization of X's future knowledge, as in (3b), X may have only an indefinite characterization of the choice specified by (5). In other words, X will know *abstractly*, but not *specifically*, what to do next. Only when the next stage of deliberation arrives, and X perceives key new information, will X be able to select a specific action.

These considerations generalize naturally to multiple steps of deliberation. At each stage, the agent must be able to choose a suitable next action based on the information then available (Moore 1985; Morgenstern 1987; Davis 1994; Stone 1998). For this paper, we will need two cycles at most. So, we will treat a one-step intention as a proof of (5), and a two-step intention as a proof of (6).

(6) $\exists a[\mathrm{X}](x \text{ DOES } a \text{ THEN } \exists a'[\mathrm{X}](x \text{ DOES } a' \text{ THEN } [\mathrm{X}]G))$

Proof-terms for Intentions

By this analysis, proof-terms that describe deliberation must track reasoning about time, reasoning about knowledge, and reasoning about indefinite information. Temporal inferences provide a basic constituent of intention representation. I formalize these inferences with recursive terms composed from symbols for agents, actions and propositions using the symbols `finish` and `step` defined in (7).

(7) a `(finish` X G`)` a demonstration that hypothesizes that agent X knows that goal G now holds, and thereby establishes that the intention has been carried through successfully.
b `(step` X A C I`)` a demonstration that hypothesizes that conditions C hold at the present moment, and thereby shows that if agent X does action A now, the right conditions will obtain afterwards to support intention I. Thus, if I is a proof of P, this is a proof of x DOES a THEN P.

The `step` structures defined in (7b) simply summarize inferences about the effects of action, so that a detailed characterization of these inferences may be carried out in any convenient formalism. A purely logical approach could check these inferences by modus ponens from axioms describing change and persistence, perhaps in the style of (Schubert 1990). I have chosen to check them by simulation, using operational specifications of actions and their effects in the style of STRIPS (Fikes and Nilsson 1971).

Modal inferences, meanwhile, are represented as in (8).

(8) `(know X C I)` a demonstration that hypothesizes that conditions C hold at present, where each condition of C must take the form $[\mathrm{x}]Q$, and thereby shows that the right conditions obtain to support intention I. If I is a proof of P then, this is a proof of $[\mathrm{x}]P$.

This corresponds to a structural approach to sequent calculus for modal logic, in which each sequent describes a particular modal context (Fitting 1983). Inferences such as (8) mark a transition from one modal context to another by allowing only formulas of a restricted syntactic form to be taken into account. By requiring here simply that the right conditions must obtain to support I, this construction also implements any additional axiom schemes (such as veridicality and positive introspection) needed to characterize knowledge for planning (Moore 1985).

Since (5) and (6) call for the proof of existential statements, I add the constructor in (9).

(9) `(find A I)` a demonstration of $\exists a.Pa$, using action A as a witness, and constructed from an intention I establishing $P(A)$.

The structure of formulas (5) and (6) now determines the way operators from (7), (8) and (9) can fit together in an intention representation. These operators must be nested: for each cycle we find first `find` A corresponding to the quantifier $\exists a$, then `know` X corresponding to the operator $[\mathrm{x}]$, then `step` X A corresponding to x DOES a THEN; at the innermost level, we have `finish`.

Finally, other reasoning operations make it possible to construct intentions that use indefinite information:

(10)a `(whatever Df If)` a demonstration that appeals to the fact that some object o satisfies the predicate `Df`, and then supplies a specific demonstration (`If` o) by applying the function `If` to o, whatever o may be. If `If` o is a proof of P, this is again a proof of P.

b `(oneof Ds Is)` a demonstration that appeals to the fact that one of the conditions spelled out in the list `Ds` must be true, and then supplies a specific demonstration, the nth element of `Is`, to suit that case where the nth element of `Ds` is true. Again, if each element of `Is` is a proof of P, this is also a proof of P.

These operations do not change the content of an intention—complex intentions correspond to proofs of the same formulas—but rather introduce ambiguities into the intention. The content of intentions, as spelled out in (5), and the corresponding reasoning about knowledge, as formalized by (8), ensures that those ambiguities are resolved by the time an agent needs to make a decision.

(11) shows an intention representation that combines all of these operators. We can read it informally as a description of a course of action appropriate when agent X knows that there is some food in the refrigerator (`food`), and desires to be relieved of hunger (`full`). X looks in the refrigerator (`look`), thereby discovering a particular item of food there (`f`). Thereafter, X eats that item (`eat f`).

(11)
```
(find look
 (know x [k x(sm f\[food f])]
  (whatever (f\[food f])
   (f\(step x look [food f]
       (find (eat f)
        (know x [k x (food f)]
         (step x (eat f) [food f]
          (finish x [full])))))))))
```

More precisely, we can regard (11) as formalizing an inference that establishes the instance of (6) in (12).

(12) $\exists a[\mathrm{x}](x$ DOES a THEN $\exists a'[\mathrm{x}](x$ DOES a' THEN $[\mathrm{x}]\mathit{full}))$

We want to show that X can be full after two feasible choices. The first choice is to look; we consider X's knowledge at this stage, namely that there is some food. For the purposes of argument, call the food f, and consider what happens when X looks: X will know that f is food. X's second choice is now to eat f. Again, we consider X's knowledge at this stage, namely that f is food, and consider what happens when X eats f: X will feel full, and know it. This proves (12).

Using Intention Representations

Earlier, I claimed that by representing intentions as inferences about deliberation that an agent is committed to, we can reconcile the use of intentions in deliberation with the force of attributions of intention. I can now support this claim with reference to the rules of (7)–(10) and specific instances such as (11).

We saw earlier that the intentions that we ascribe to other agents must encapsulate a complex web of goals and beliefs. At the highest level, we can use the relations on intentions of (13) to describe this complex structure.

(13)a `(expectation I E)` E gives one way I's expectations could be true.

b `(nextAct I E A)` when expectations E hold, I gives a reason to do A now.

c `(nextIntention I E N)` when expectations E hold, I suggests trying N next.

d `(goals I G)` I envisages outcome G.

Our proof-theoretic representation naturally supports the implementation of these relations. In particular, the expectations are the hypotheses that the proof appeals to; the next action is the first act hypothesized in the proof, represented in the outermost `find` term; the next intention is a subproof that considers the agent's next stage of execution and action, represented by `find` terms at one level of embedding; the goal is the condition that the proof ultimately establishes, represented in innermost `finish` terms.

These relations then provide a natural vocabulary of constraints on intentions with which to specify deliberation with intentions and attribution of intentions. With these resources, for example, we can specify the reasoning of an agent that sticks to its intention as the clause in (14).

(14)
```
deliberate B I A I' :-
  expectation I E,
  met E B,
  nextAct I E A,
  nextIntention I E I'.
```

The agent considers its current beliefs `B` and present intention `I`. The agent first ensures that some `expectation` `E` of `I` is `met` according to `B`; in this case, the intention spells out action `A` to do now and intention `I'` to follow next. The agent simply selects action `A` and commits to intention `I'`.

Let's step through an example: agent *X* acting on the intention of (11). Initially, this intention depends on the expectation that *X* knows there is some food; let us assume that this expectation is met, so that *X*'s deliberation in this cycle can proceed normally as specified in (14). Then *X* will decide to look now. For the next cycle, *X* will commit to an abstract intention derived from (11), as in (15).

(15)
```
(find (eat F)
 (know x [k x (food F)]
  (step x (eat F) [food F]
   (finish x [full])))))))))
```

In this transition we have used the bound-variable representation of individuals assumed for the sake of argument, and introduced a new logic variable `F` to leave open the object that *X* will eat at the next stage.

We now move to the second stage of deliberation. Suppose that in fact, as part of *X*'s perceptions in the second cycle, *X* identifies a particular piece of cheesecake *c* in the refrigerator. Meanwhile, the intention of (15) involves a general expectation of the form `k x (food F)`. In showing that *X*'s information meets the expectations of (15), we can use the fact that *X* knows that *c* is food by instantiating `F` to `c`. This is just the usual process of unification in logic-programming proof search. Given this instantiation, the intention of (15) specifies `eat c` as the next action,and `finish x [full]` as the next intention.

In effect, then, in this example, the intention has not only focused *X*'s attention and deliberation on the circumstances requires for the plan of action to succeed, but simultaneously determined a specific choice for *X* to carry the intention through based on new information.

Collaborative Agency

The intention representations defined in (7), (8) and (10) and illustrated in (11) and (15) account for the action of a single agent acting in isolation. In fact, however, I will adopt the same representations for *collaborative* intentions as well.

I assume that a fixed group of agents are committed to the ongoing collaboration; I will refer to these agents as the team *T* and, by extension, I will use [T] to describe the knowledge that all members of the team share. (Actually, I will be most interested in dialogues between two agents.) To represent collaborative intentions for *T*, I simply allow any of the agents in *T* to perform any step of action in the intention. However, in each cycle of execution, the intention must restrict attention to the knowledge of the acting agent; thus demonstration achieved in each step of the plan continues to exhibit the find-know-step structure seen for example in (11). Each agent has the information required to act, when it is their turn to do so.

In reusing the architecture of (7)–(10), I adopt a view of collaboration as many agents acting as one. The metaphor is appealing, but as researchers such as (Tambe *et al.* 1999; Sengers 1999) have observed, pursuing it requires retooling general models of agency to make explicit the trust that each agent places in the team, and the responsibilities that each agent owes it. In particular, each agent must reason by *intention recognition* to interpret its teammates' actions, and to allow its teammates to interpret its own actions, in the context of the evolving collaboration.

To interpret an agent's action, another agent must reconstruct the mental representations and steps of deliberation in which the action originates. This inferred intention must provide a consistent representation of choice, action and effect, as mapped out in (7)–(10). The inferred intention must involve actions consistent with what the recognizer has observed, and must set out the expectations that are consistent with the recognizer's model of the actor's beliefs. More generally, the goals of the inferred intention must be consistent with the recognizer's model of the actor's desires. When the recognizer has only partial information about the actor, intention recognition becomes explosive, because the recognizer may adopt new assumptions about the actor as a possible explanation for the actor's actions.

Collaboration, by contrast, allows for *intended* recognition, a potentially much simpler reasoning task. Agent and observer are on the same team, and so coordinate not only so that each action achieves the effects in the real world to which the team is committed, but also so that the team as a whole understands the status of each action (Cohen and Levesque 1991). By appealing to *common ground* (Clark and Marshall 1981) as a public record of the state of the team's collaboration, we can understand collaborative intention recognition as a constraint-satisfaction problem in which agents match as much information as possible against this shared record.

To illustrate how intention representations can support collaboration, I return to the culinary conceit introduced in (11). Let us now imagine that two agents *X* and *Y* are collaborating to prepare mashed potatoes. In the collaboration, agent *X* peels each potato and passes it to agent *Y*. I represent this action as `peel P`, depending on the potato `P`, and assume that the successful completion of this action makes it clear to both agents that the potato is `clean`. Then, agent *Y* cuts the potato into chunks of a consistent size, so as to ensure that all the potatoes cook evenly. In other words, although cleaning the potato may substantially change the size of the potato, agent *Y* will know from perceiving the potato what pattern of cuts `z` fits the potato; I represent this knowledge as a fact `k y (cutp P z)`. Thus, *Y* knows that performing the action `cut P z` of cutting the potato as specified by `z` is the right step to make the potato `ready` for boiling.

The representation in (16) formalizes this plan of joint action as an inference.

(16)
```
(find (peel P)
 (know x [k t (potato P)]
  (step x (peel P) [potato P]
   (whatever (z\[k y (cutp P z)])
    (z\(find (cut P z)
        (know y [k y (cutp P z),
                 k t (clean P)]
         (step y (cut P z)
```

```
              [cutp P z, clean P]
          (finish t [ready P]))))
      )))))
```

First X peels, then Y cuts (as decided after the peeling), and finally the potato is made ready. The potato is schematized throughout by a logic variable `P` so that this representation can serve as a resource for the two agents' collaboration.

Thus, suppose X peels potato `p8` in a first stage of collaborative decision-making and action. Relying on (16) as a potential expectation about the interaction, any agent on the team, relying only on information in the common ground, can recognize X's intention in peeling `p8` as an instance of (16) with `P=p8`. In particular, by such inference X can anticipate that peeling `p8` will make an unambiguous contribution to the collaboration. Meanwhile, as Y interprets X's action, Y can commit to the same intention and proceed with the next stage of decision-making and action for the ongoing task.

In this next stage, each agent on the team starts with the intention in (17).

```
(17)  (find (cut p8 Z)
       (know y [k y (cutp p8 Z),
                k t (clean p8)]
        (step y (cut p8 Z)
              [cutp p8 Z, clean p8]
         (finish t [ready p8])))))
```

This structure reflects the instantiation of `P` to `p8` from (16) but now includes a logic variable `Z` to schematize the pattern of cuts that is yet to be decided.

It falls to Y to act on this intention. As with (15) this means checking that the expectations laid out in the intention are met; indeed, as with (15), a side-effect of this check in a logic-programming setting is the instantiation of `Z` to a specific value, `thirds` let us say. So agent Y will select the specific action `cut p8 thirds`, governed by a specific intention as in (18).

```
(18)  (find (cut p8 thirds)
       (know y [k y (cutp p8 thirds),
                k t (clean p8)]
        (step y (cut p8 thirds)
              [cutp p8 thirds, clean p8]
         (finish t [ready p8])))))
```

Now Y's action must be interpreted. Drawing on the common ground, the team combines the expectation (17) with the observed action `cut p8 thirds`. Taking a constraint-satisfaction perspective on intention recognition in collaborative process, the team can assume that Y's action does indeed fit the expectation of (17) under `Z=thirds` and so reconstruct (18) using shared information. Thus, after the second cycle, all agents recognize Y's judgment that `thirds` is the right pattern of cuts for `p8`, and all are prepared to jointly assess the results of the collaboration—that `p8` is in fact now `ready` to cook.

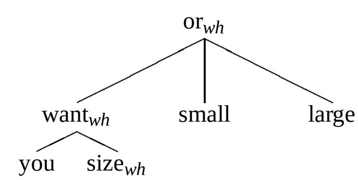
Language Use as Collaborative Agency

Our representation of collaborative intention and agency now enables us to pursue a formal parallel between participation in dialogue and the kind of real-world collaboration illustrated in (16)–(18).

Consider (1) again, for example. In (1a), we recognize a specific intention, in which the initiating speaker X asks the question—*what size do you want: small or large?*—and then the respondent Y draws on the established words to formulate an utterance that provides the answer (in this case *large*). For now, let's call this intention i_1. In understanding the question, the respondent Y recognizes i_1: Y knows that X intends a response. In choosing to offer an answer in reply as in (1), Y simply cooperatively adopts this plan and follows it through. This reasoning parallels that for (16)–(18).

To pursue the parallel further, I now develop a suitable model of grammatical structure and linguistic knowledge, with which we can represent intentions such as i_1 within the schemas provided by (7)–(10). I assume dependency representations of syntactic structure which analyze each utterance as a specified combination of meaningful elements. The derivation trees of tree-adjoining grammar provide such a representation (Joshi *et al.* 1975; Vijay-Shanker 1987; Schabes 1990). For (1a) the tree in (19) is suggestive of such dependency representations.

(19)



To specify linguistic actions, I represent such trees as logic programming terms, as in (20).

```
(20)  or_wh (want_wh you size_wh) small large
```

Meanwhile, I assume that linguistic knowledge includes a compositional semantics that defines a relation `content U M` true when `M` represents the linguistic meaning of syntactic object `U`. For X's utterance (1a), for example, we require the instances of this relation given in (21).

```
(21)a content (want_wh you size_wh)
              k\(and (want y k) (is_size k))
    b content small is_small
    c content large is_large
```

(Strictly, for λProlog, the type of `M` also matters, but in this example `M` is always a property.)

This content determines the effects that we understand actions such as (20) to have. We associate (20) with the *presupposition* in (22a) and the *contribution* in (22b).

```
(22)a [k t (is_small K1), k t (is_large K2)]
    b k t (oq k\(and (want y k) (is_size k)))
```

When its presupposition is true, (20) adds the contribution to the common ground. That is, when there are two kinds in the context, one small and one large, (20) introduces the *open question* (`oq`) of which size Y wants.

Figure 1 maps out the intention i_1 in the representation of (7)–(10) using these assumptions about linguistic actions and their effects. The *schema* of Figure 1 maps out the inferences involved in coordinating to add information to the discourse using a *wh*-question-answer pair with two alternatives explicitly provided; variables in the schema are subject to the specified *constraints*. For i_1, X uses the schema under the specified *instances*.

```
schema: (find (or_wh SWH XP1 XP2)
         (know x [k t (P1 K1), k t (P2 K2), k x (or [k y (Ans K1), k y (Ans K2)])]
          (step x (or_wh SWH XP1 XP2)
              [k t (P1 K1), k t (P2 K2), k t (or [k y (Ans K1), k y (Ans K2)])]
           (oneof [k y (Ans K1), k y (Ans K2)]
            [(find XP1
              (know y [k t (oq Ans), k t (P1 K1), k y (Ans K1)]
               (step y XP1 [k t (oq Ans), k t (P1 K1), k y (Ans K1)]
                (finish t [k t (Ans K1)])))),
             (find XP2
              (know y [k t (oq Ans), k t (P2 K2), k y (Ans K2)]
               (step y XP2 [k t (oq Ans), k t (P2 K2), k y (Ans K2)]
                (finish t [k t (Ans K2)]))))]))))

constraints: content SWH Ans   instances: SWH=(want_wh you size_wh) Ans=k\(and (want y k) (is_size k))
             content XP1 P1               XP1=small P1=is_small K1=k1
             content XP2 P2               XP2=large P2=is_large K2=k2
```

Figure 1: Communicative intention schema and instance for (1a).

The reasoning in Figure 1 parallels (16). *X* chooses the question, knowing that the presupposition is met and that *Y* knows the answer. So the question makes its contribution, and we consider separately the different alternatives for what *Y* knows of the answer. In each case, *Y* draws on *X*'s description and exploits the effects of *X*'s question to contribute the answer to the context.

The formal representations of Figure 1 contribute to an account of the dynamics of (1) that proceeds along the following lines. To recognize i_1 on shared information, conversationalists must recognize that the schema applies, instantiate the schema for the observed action, and instantiate any remaining variables by matching the expectations and goals of the plan against the common ground. We may assume this offers *Y* no difficulty, and that *Y* then commits to i_1.

At the next stage of deliberation, *Y*'s turn, *Y* is committed to a disjunctive intention whose expectations can be met in one of two ways: *Y* wants the small size or *Y* wants the large size. In this case, by matching the expectations, *Y* selects the latter, deriving a specific intention for an utterance of *large* that can, of course, be recognized as intended. Afterwards, then, both *X* and *Y* are prepared to jointly assess the results of the collaboration: it's settled that *Y* wants the large size.

Assessment and Prospects

In modeling intentions as a *resource* for collaborative deliberation this way, the framework realized here distinguishes two kinds of reasoning about action in collaboration. An underlying *collaborative process* manages steps of coordination, plan recognition and (in language use) such functions as turn-taking, acknowledgment and grounding. Specific representations of collaborative intention presuppose and factor out this underlying process in characterizing agents' choice and action towards team goals. Because of this, inferences such as i_1 can exhibit full linguistic and logical detail while remaining substantially simpler than other pragmatic formalizations, such as those based on speech-act theory (Searle 1969; Cohen and Perrault 1979; Allen and Perrault 1980). The simplicity is essential for modeling phenomena—for starters, entrainment—which depend on the pragmatic consequences of syntactic and semantic choices.

As formalized in Figure 1, intentions like i_1 constitute comprehensive pragmatic representations of utterance interpretation. They specify the syntax and semantics of the utterance and link this specification by inference into a description of what a speaker might do with the utterance in context. In my prototype implementation, these detailed representations of communicative intentions serves as the common object of diverse processes in conversational reasoning—from interpretation through dialogue management to utterance generation. I hope that the analysis of (1) suggests the elegance that this design affords.

References

J. F. Allen and C. R. Perrault. Analyzing intention in utterances. *Artificial Intelligence*, 15:143–178, 1980.

M. E. Bratman, D. J. Israel, and M. E. Pollack. Plans and resource-bounded practical reasoning. *Computational Intelligence*, 4:349–355, 1988.

M. E. Bratman. *Intention, Plans, and Practical Reason*. Harvard University Press, Cambrdige, MA, 1987.

S. E. Brennan and H. H. Clark. Conceptual pacts and lexical choice in conversation. *Journal of Experimental Psychology: Learning, Memory and Cognition*, 22(6):1482–1493, 1996.

S. E. Brennan. *Seeking and Providing Evidence for Mutual Understanding*. PhD thesis, Stanford University, 1990.

S. E. Brennan. Lexical entrainment in spontaneous dialog. In *International Symposium on Spoken Dialogue*, pages 41–44, 1996.

J. Cassell, D. McNeill, and K. E. McCullough. Speech-gesture mismatches: evidence for one underlying representation of linguistic and nonlinguistic information. *Pragmatics and Cognition*, 6(2), 1999.

J. Cassell. Nudge nudge wink wink: Elements of face-to-face conversation for embodied conversational agents. In J. Cassell, J. Sullivan, S. Prevost, and E. Churchill, editors, *Embodied Conversational Agents*, pages 1–28. MIT Press, Cambridge, MA, 2000.

B. A. Cheikes. *Planning Responses from High-Level Goals: Adopting the Respondent's Perspective in Cooperative Response Generation*. PhD thesis, University of Pennsylvania, Philadelphia, 1991.

H. H. Clark and C. R. Marshall. Definite reference and mutual knowledge. In A. K. Joshi, B. L. Webber, and I. Sag, editors, *Elements of Discourse Understanding*, pages 10–63. Cambridge University Press, Cambridge, 1981.

H. H. Clark and E. F. Schaefer. Contributing to discourse. *Cognitive Science*, 13:259–294, 1989.

H. H. Clark and D. Wilkes-Gibbs. Referring as a collaborative process. *Cognition*, 22:1–39, 1986.

H. H. Clark. *Using Language*. Cambridge University Press, Cambridge, UK, 1996.

P. R. Cohen and H. J. Levesque. Intention is choice with commitment. *Artificial Intelligence*, 42:213–261, 1990.

P. R. Cohen and H. J. Levesque. Teamwork. *Nous*, 24(4):487–512, 1991.

P. R. Cohen and C. R. Perrault. Elements of a plan-based theory of speech acts. *Cognitive Science*, 3(3):177–212, 1979.

E. Davis. Knowledge preconditions for plans. *Journal of Logic and Computation*, 4(5):721–766, 1994.

R. Fagin, J. Y. Halpern, Y. Moses, and M. Y. Vardi. *Reasoning About Knowledge*. MIT Press, Cambridge MA, 1995.

G. Ferguson and J. F. Allen. Arguing about plans: Plan representation and reasoning for mixed-initiative planning. In K. Hammond, editor, *Proceedings of the Second International Conference on A.I. Planning Systems*, pages 43–48, 1994.

R. E. Fikes and N. J. Nilsson. STRIPS: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, 2:189–208, 1971.

M. Fitting and R. L. Mendelsohn. *First-order Modal Logic*, volume 277 of *Synthese Library*. Kluwer, Dordrecht, 1998.

M. Fitting. *Proof Methods for Modal and Intuitionistic Logics*, volume 169 of *Synthese Library*. D. Reidel, Dordrecht, 1983.

J. Gallier. Constructive logics. I. A tutorial on proof systems and typed λ-calculi. *Theoretical Computer Science*, 110(2):249–339, 1993.

H. P. Grice. Meaning. *The Philosophical Review*, 66(3):377–388, 1957.

J. Y. Halpern and Y. Moses. A guide to the modal logics of knowledge and belief: preliminary draft. In *Proceedings of the Ninth IJCAI*, pages 480–490, 1985.

J. Hanna and M. Tanenhaus. The use of perspective during referential interpretation. In *CUNY Sentence Processing Conference*, page 93, 2001.

J. Hintikka. Semantics for propositional attitudes. In Linsky, editor, *Reference and Modality*, pages 145–167. Oxford, 1971.

A. K. Joshi, L. Levy, and M. Takahashi. Tree adjunct grammars. *Journal of the Computer and System Sciences*, 10:136–163, 1975.

H. Kamp and U. Reyle. *From Discourse to Logic: Introduction to Modeltheoretic Semantics of Natural Language, Formal Logic and Discourse Representation Theory*. Kluwer, Boston, 1993.

D. Lewis. Scorekeeping in a language game. In *Semantics from Different Points of View*, pages 172–187. Springer Verlag, Berlin, 1979.

B. F. Malle and J. Knobe. The folk concept of intentionality. *Journal of Personality and Social Psychology*, 33:101–121, 1997.

D. Miller. A logic programming language with lambda-abstraction, function variables, and simple unification. *Journal of Logic and Computation*, 1(4):497–536, 1991.

R. C. Moore. A formal theory of knowledge and action. In Jerry R. Hobbs and Robert C. Moore, editors, *Formal Theories of the Commonsense World*, pages 319–358. Ablex, Norwood NJ, 1985.

L. Morgenstern. Knowledge preconditions for actions and plans. In *Proceedings of the 10th International Joint Conference on Artificial Intelligence*, pages 867–874, Milan Italy, 1987.

R. Muskens. Combining Montague semantics and discourse representation. *Linguistics and Philosophy*, 19(2):143–186, 1996.

G. Nadathur and D. Miller. Higher-order logic programming. In D. Gabbay, C. J. Hogger, and J. A. Robinson, editors, *Handbook of Logics for Artificial Intelligence and Logic Programming*, pages 499–590. Oxford, 1998.

F. C. N. Pereira and S. M. Shieber. *Prolog and Natural Language Analysis*. CSLI, Stanford CA, 1987.

F. Pfenning. Structural cut elimination I. Intuitionistic and classical logic. *Information and Computation*, 157(1/2):84–141, 2000.

M. E. Pollack. Plans as complex mental attitudes. In P. R. Cohen, J. Morgan, and M. E. Pollack, editors, *Intentions in Plans and Communication*, pages 77–103. MIT Press, Cambridge MA, 1990.

M. E. Pollack. The uses of plans. *Artificial Intelligence*, 57:43–68, 1992.

A. S. Rao and M. P. Georgeff. An abstract architecture for rational agents. In *Proceedings of Knowledge Representation and Reasoning*, pages 439–449, 1992.

Y. Schabes. *Mathematical and Computational Aspects of Lexicalized Grammars*. PhD thesis, Computer Science Department, University of Pennsylvania, 1990.

L. K. Schubert. Monotonic solution of the frame problem in the situation calculus: an efficient method for worlds with fully specified actions. In H. E. Kyburg, R. P. Loui, and G. N. Carlson, editors, *Knowledge Representation and Defeasible Reasoning*, pages 23–67. Kluwer, Boston, 1990.

J. R. Searle. *Speech Acts: An Essay in the Philosophy of Language*. Cambridge University Press, Cambridge, 1969.

P. Sengers. Designing comprehensible agents. In *Proceedings of IJCAI*, 1999.

M. Stone. Abductive planning with sensing. In *AAAI*, pages 631–636, Madison, WI, 1998.

M. Tambe, J. Adibi, Y. Al-Onaizan, A. Erdem, G. A. Kaminka, S. C. Marsella, and I. Muslea. Building agent teams using an explicit teamwork model and learning. *Artificial Intelligence*, 110:215–240, 1999.

R. H. Thomason. Accommodation, meaning and implicature. In P. R. Cohen, J. Morgan, and M. E. Pollack, editors, *Intentions in Communication*, pages 325–363. MIT Press, Cambridge, MA, 1990.

R. van der Sandt. Presupposition projection as anaphora resolution. *Journal of Semantics*, 9(2):333–377, 1992.

K. Vijay-Shanker. *A Study of Tree Adjoining Grammars*. PhD thesis, Department of Computer and Information Science, University of Pennsylvania, 1987.