MITSUBISHI ELECTRIC RESEARCH LABORATORIES
http://www.merl.com

Generating, Recognizing and Communicating Intentions in Human-Computer Collaboration

Charles Rich and Candace L. Sidner

TR2007-010   February 2007

Abstract

We describe the semantics of the plan representation in Collagen and how it supports the generation, recognition and communication of intentions in human-computer collaboration. This integrated architecture is made possible in large part because of the foundations of the representation in collaborative discourse theory.

AAAI Spring Symposium 2007

This work may not be copied or reproduced in whole or in part for any commercial purpose. Permission to copy in whole or in part without payment of fee is granted for nonprofit educational and research purposes provided that all such whole or partial copies include the following: a notice that such copying is by permission of Mitsubishi Electric Research Laboratories, Inc.; an acknowledgment of the authors and individual contributions to the work; and all applicable portions of the copyright notice. Copying, reproduction, or republishing for any other purpose shall require a license with payment of fee to Mitsubishi Electric Research Laboratories, Inc. All rights reserved.

Copyright © Mitsubishi Electric Research Laboratories, Inc., 2007
201 Broadway, Cambridge, Massachusetts 02139

Generating, Recognizing and Communicating Intentions in Human-Computer Collaboration

Charles Rich
Mitsubishi Electric Research Laboratories
201 Broadway
Cambridge, MA, 02139, USA
rich@merl.com

Candace L. Sidner
BAE Systems Advanced Information Technologies
6 New England Executive Park
Burlington, MA 01803, USA
candy.sidner@baesystems.com

Abstract

We describe the semantics of the plan representation in Collagen and how it supports the generation, recognition and communication of intentions in human-computer collaboration. This integrated architecture is made possible in large part because of the foundations of the representation in collaborative discourse theory.

Introduction

Collagen is a system for building collaborative interface agents based on the collaborative discourse theory of Grosz, Sidner, et al. (Grosz & Sidner 1986; 1990; Grosz & Kraus 1996; Lochbaum 1998). Collaborative interface agents are computer programs that cooperate with humans through the use of action and communication (in natural or artificial language) to achieve shared goals. Collagen has been used both by the authors and by others outside our laboratory to develop more than a dozen such prototype agents (both purely software and physically embodied (Sidner *et al.* 2005)) for a wide variety of applications (Rich, Sidner, & Lesh 2001; Rich & Sidner 2007).

Collaborative discourse theory is comprised of three interrelated components: intentional, attentional, and linguistic. All three of these are implemented in Collagen. Very roughly speaking, the intentional component is implemented by plan trees, the attentional component is implemented by a focus stack, and the lingustic component is implemented by an artificial negotiation language.

In this paper, we focus on Collagen's intentional component, specifically how plan trees support the generation, recognition and communication of intentions. The main advantage of this architecture is that it facilitates the fine-grained interleaving of these three processes required for collaboration. Having a single representation with a clear semantics has, we believe, led us to a simpler and more consistent implementation than would otherwise be possible.

To our knowledge, this integrated architecture is unique to Collagen among generic tools for building intelligent agents.[1] Many intelligent tutoring systems (tutoring is a kind of collaboration) use a single plan-like representation both for generating agent (tutor) intentions and recognizing user (student) intentions. However, the communication process (tutorial dialogue) in these systems has not generally been based on a semantic foundation in terms of intentions. Our joint work with Rickel (2002) was directed toward bringing this sort of dialogue semantics into intelligent tutoring.

Several recent collaborative agent building systems, such as STAPLE (Kumar & Cohen 2004), Artimis (Sadek, Bretier, & Panaget 1997), and Rochester's collaborative dialogue agent (Ferguson & Allen 2007), use the semantics of the agent's intentional represention for interpretation of user utterances, thereby integrating communication and plan generation. However, none of these includes a powerful plan recognition component such as Collagen's, i.e., one that can do interpolation (see discussion of plan recognition below).

Plans

Before describing how Collagen's intentional representation is used, we first need to describe its semantics in terms of collaborative discourse theory.

As Pollack (1990) has pointed out, the term "plan" has historically been used confusingly to refer to two different concepts: *knowledge* about how to achieve certain goals (e.g., by decomposing them into subgoals), and structures of *intentions* to achieve certain goals and subgoals. In Collagen, we follow Pollack's suggested terminology and use the term "recipes" for the former concept (see next section), reserving the term "plan" for the latter concept.

Collagen's plan trees (technically, directed acyclic graphs, since we allow a node to have multiple parents) are a simplified implementation of the SharedPlan representation of collaborative discourse theory. An agent's intentional state (part of its overall cognitive state) may contain multiple plan trees, corresponding to multiple toplevel goals.

Each node in a plan tree (see example in Figure 1) is an instance of an act type, which means that it may provide values for some or all of the parameters defined by the act type, and for the act's participants. We treat utterances as actions in Collagen. Thus nodes in the plan tree correspond to both utterances and "physical" actions (manipulations) in the world.

The semantics of a plan tree node is either an individual (agent private) plan or a Partial SharedPlan (between the user and the agent) with respect to that act, as further described below. SharedPlans include a rich framework for modeling groups of participants in acts. In Collagen, how-

[1] Though part of the goal of this workshop is to find out for sure!

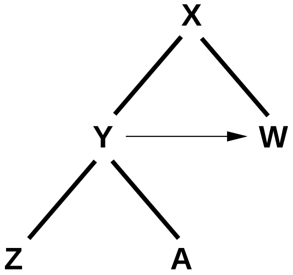


Figure 1: An example plan tree.

ever, the only possible values for the participants in an act are {*agent*}, {*user*}, and {*agent,user*}.[2] If the participants value for a node (act) is {*agent*}, then it represents an intention to perform the act; otherwise the node represents an "intention that" (Grosz & Kraus 1993) the act be achieved.[3]

In the usual logical notation, where act type A is formalized as a predicate whose first argument is the participants, the following are the intentions represented by an instance of A in the plan tree, depending on the participants value:

Intend(agent, A({agent}))
IntendThat(agent, A({agent,user}))
IntendThat(agent, A({user}))
IntendThat(agent, A(?))

This node and its children (see below) represent (at least) an individual plan on the part of the agent to achieve A.

In addition to the intentions above, a Partial SharedPlan (i.e., non-private plan) for A requires roughly the following mutual beliefs (MB) between the agent and user, depending again on the participants in A: [4]

MB(Intend(agent, A({agent})))
MB(IntendThat(agent, A({user})))
MB(IntendThat(user, A({agent,user})))
MB(IntendThat(agent, A({agent,user})))

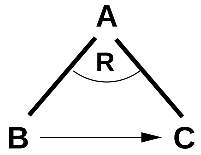
These mutual beliefs arise either from the process of plan recognition or communication, both of which add belief annotations to nodes in the plan tree, as described below.

Because Collagen has mostly been used thus far to develop "compliant" agents, i.e., agents which always accept user goals as their own, it does not currently provide a plan node belief annotation for the case of user intentions which are not also agent intentions. For example, we cannot currently represent just the agent belief

Believe(agent, IntendThat(user, A({agent})))

without the corresponding agent intention

Intend(agent, A({agent})).



```
recipe R achieves A {
   step B b;
   step C c:
   constraints {
      precedes(b, c);
      achieves.param1 == b.param1;
      b.param2 == c.param1;
      c.param2 == achieves.param2;
   }
}
```

Figure 2: An example recipe.

However, it is a small extension to support this belief (basically, a matter of adding a boolean property to plan nodes). Furthermore, this extension will be required when we implement the rest of Sidner's artificial negotiation language (see Communication section below) to support agents that can negotiate actions.

The arcs in a plan tree represent the SharedPlan relationship *directly contributes*, in which the act represented by the child (subplan) node directly contributes to the act represented by the parent. Examples of this relationship include:

- the subplan results in a choice of recipe for the parent
- the subplan is an instance of a step in the recipe for the parent (see recipe application description in next section)
- the subplan results in assigning a value to a parameter of the parent
- the subplan results in assigning the parent's participants

As in SharedPlans, Collagen's plan representation also specifies a partial order between the subplans of each node (illustrated by the horizontal arrow in Figure 1). Subplans may thus be totally ordered, unordered, or any consistent partial ordering in between.

Finally, plan trees in Collagen are annotated with propositional constraints between the parameters of the act instances represented by the nodes. The most important of these constraint types is equality, which propagates values both "up" and "down" the tree (down from generation and up from recognition—see more below). Reasoning with these constraints is implemented by a boolean truth maintenance system (a TMS with integrated unit propositional resolution—a fast but incomplete decision procedure for boolean logic) integrated with a complete decision procedure for equality.

Recipes

Following Pollack and SharedPlans, recipes in Collagen are essentially goal decomposition rules. Recipes are used both in the generation and recognition of collaborative behavior, as described in the next two sections. In both processes, the

[2] We have recently experimented with extending this to three-party collaborations.

[3] SharedPlan theory does not allow direct intentions with respect to an act that involves other participants; instead it defines a concept of "intend that" for such intentions.

[4] These mutual beliefs and the intentions above should all be wrapped in "Believe(agent, ...)," since they are part of agent's cognitive state; we have omitted this for ease of reading.

key operation is to *apply* a recipe to a node in a plan tree. (In addition, as with any kind of knowledge, recipes can be explained, learned, etc.)

One way to think of a recipe is as a plan tree "template." When a recipe is applied to a plan node of the required type, it adds children to that node with a specified ordering (if any) between the children and propositional constraints between the children and typically including the parent. Figure 2 shows an example of a recipe, both in diagrammatic form and in the Java-based source code notation used by Collagen. This recipe decomposes an instance of act type A into two totally ordered steps of type B and C, with the result parameter of the first step becoming an input parameter to the second step.

The procedural knowledge of a collaborative agent implemented using Collagen consists predominantly of a library of such recipes, including possibly more than one recipe for each act type.

Both plans and recipes are drawn as trees. However, it is very important to remember Pollack's distinction and not confuse the two. If the recipe in Figure 2 were applied to the A node in Figure 1 the resulting plan tree would, in diagrammatic form, appear (see Figure 3) as if the subtree below A in Figure 2 were appended to the fringe of tree in Figure 1. From a semantic point of view, however, this amounts to adding new intentions and/or beliefs to the agent's cognitive state corresponding to the new instances of B and C.

Plan Generation

Plan generation is fundamentally the process of adding subplans to plan trees which, as mentioned above, corresponds to the agent adding new intentions and/or beliefs to its cognitive state. Unlike plan recognition and communication, discussed in the next sections, plan generation only extends the agent's current individual plan; it does not result in new beliefs about the *user's* intentions. For example, if the agent generates a new subplan for the user to perform A, the semantics of the new node is:

IntendThat(agent, A({user}))

Note that this is only an agent intention. In the absence of other information, not only may the user not intend at this point to perform A, the user might not even know that A exists! In a typical collaboration, the agent will eventually communicate its intention to the user by an utterance with ProposeShould semantics (see Communication section).

In Collagen, plan generation currently occurs by several different methods, with an additional one expected in the future.

The most important method of plan generation is via recipe application. For example, Figure 3 shows the result of applying the recipe in Figure 2 to the A node in the plan in Figure 1. Notice that the choice of recipe R is noted in the resulting plan. How to choose when to expand a given plan node, or, when there is a choice of recipes, which recipe to apply, is beyond the scope of this paper. It is worth noting, however, that the choice of recipe can itself be the topic of communication with a collaborative agent.

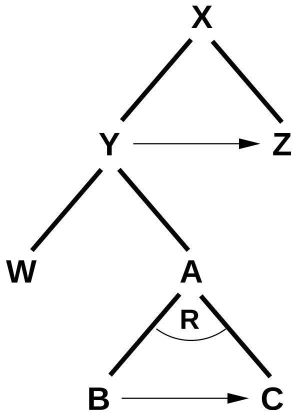


Figure 3: An example of applying a recipe.

Collagen's plan representation does not, however, require a recipe at every plan node with subplans. Subplans can also be added one-by-one, or in groups, by any other process. A boolean annotation on each plan node indicates whether or not the current subplans are complete, i.e., whether their execution is expected to achieve the parent (this annotation is set to true by recipe application).

The second most important method of plan generation in Collagen is via what we call "plug-ins." The primary purpose of plug-ins is to generate an agenda of possible actions (including utterances), which the agent might perform given its current intentional and world state. This agenda is generated by applying every plug-in (a small piece of generic or application-specific code) to every node in the current plan tree (for more details, see (Rich *et al.* 2002)). As a side effect, [5] however, a plug-in may also add one or more subplans to the node it is applied to.

Finally, as a last resort, application-specific code in the agent may add subplans based on arbitrary criteria. This method is not often used in Collagen, but is important as a practical fallback.

An important plan generation method currently missing from Collagen is first-principles planning. Unlike the other methods described above, first-principles planning requires a complete axiomitization of all the act types involved in the planning process (which is why it was not emphasized in our work to date). It is clear where this method fits into the current architecure, i.e., as yet another process which adds subplans. We hope to take an off-the-shelf partial-order planner and integrate it into Collagen in the near future.

Plan Recognition

Although plan recognition is a well-known feature of human collaboration, it has proven difficult to incorporate into practical human-computer collaboration systems due to its inherent intractability in the general case. In Collagen, we exploit

[5] There are in fact problems, e.g., for hypothetical reasoning, with allowing plug-ins to directly modify the plan tree. We have an improved, but as yet unimplemented, design in which a plug-in *returns* a set of potential subplans.

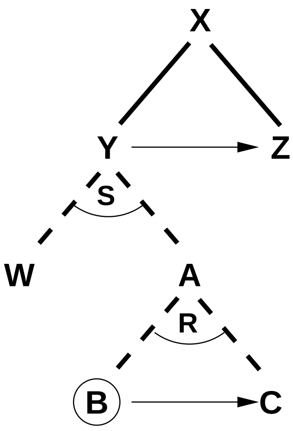


Figure 4: An example of plan recognition.

the following three properties of the collaborative setting in order to make plan recognition practical:

- the focus of attention
- the use of partially elaborated hierarchical plans
- the possibility of asking for clarification

Plan recognition extends the plan tree at the fringe, at minimum by adding a single node (representing an instance of an observed atomic act), or more generally, by interpolating a new subtree between the existing fringe of the tree and the observed act (based on the recipe library provided).

Figure 4 shows an example of plan recognition involving interpolation, in which an instance of act B is observed in the intentional state containing the plan tree XYZ, and assuming a recipe library including R from Figure 2 and another recipe S, which decomposes Y into W and A. In this example, three new nodes are added to the plan tree: instances of W, A and C.[6] Each of these nodes has a belief annotation which, for example for A, selects the following semantics:

Believe(agent, IntendThat(user, A(?)))

This belief, together with appropriate axioms for a compliant collaborative agent, provide the basic semantics for a Partial SharedPlan for A, namely:

MB(IntendThat(user, A({agent,user})))

In this example, there was exactly one possible extension to the current plan which accounted for the observed act. There can, in general, also be zero or more than one such extensions. If there is more than one, Collagen delays extending the plan until subsequent events disambiguate the possibilities. (Typically, the agent will ask a disambiguating question when the number of ambiguous acts exceeds a system parameter.) If there are no possible extensions which account for the observed act, then Collagen starts a new toplevel plan representing an unknown goal. For more details on Collagen's plan recognition algorithm, see (Lesh, Rich, & Sidner 1999).

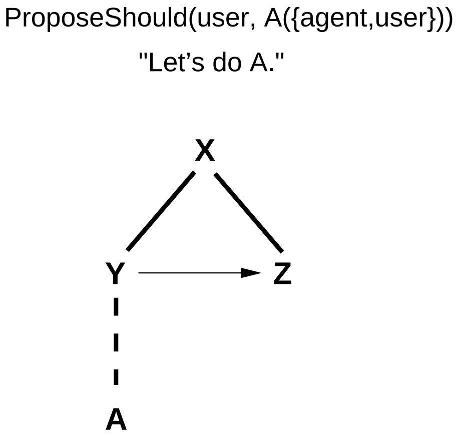


Figure 5: An example of proposing an action.

Communication

The third and final piece of this story is about how agent-user communication in Collagen extends the intentional state of the agent. The key to this is Sidner's artificial negotiation language (Sidner 1994), which specifies utterance semantics in terms of updates to the beliefs of the speaker and the hearer. Describing Sidner's complete language is beyond the scope of this paper (and beyond Collagen's current implementation). We focus here on a single utterance type, which we call ProposeShould.

Intuitively, ProposeShould is a way of communicating one's intention to perform some act. For example, the default English surface forms generated by Collagen for the three possible ProposeShould utterances by the agent with respect to act A are shown in the table below:

ProposeShould(agent, A({agent,user}))	"Let's do A."
ProposeShould(agent, A({agent}))	"I'm going to do A."
ProposeShould(agent, A({user}))	"Please do A."

Utterances such as these are how the agent converts its individual plan for A into a Partial SharedPlan plan for A. Sidner's semantics for ProposeShould (which derive from her semantics for proposals in general) allows the agent to add the following belief to its cognitive state (assuming certain conditions about the communication channel being open, etc.) after, for example, the first utterance above:

Believe(agent,
    Believe(user, IntendThat(agent, A({agent,user}))))

This is not yet enough, however, for mutual belief, which is attained only once the the user *accepts* the agent's proposal, either implicitly or explictly (e.g., by "Ok"). Applying Sidner's general semantics for acceptance of proposals, the agent may then add the following beliefs to its state:

MB(IntendThat(user, A({agent,user})))
MB(IntendThat(agent, A({agent,user})))

In this case, i.e., an agent ProposeShould, there was already a node for A in the plan tree. The result of the communication was to achieve the mutual beliefs required for a Partial SharedPlan.

[6] The observed instance B is also added to the plan tree for historical purposes.

Now consider the corresponding ProposeShould by the user, i.e., the user says "Let's do A." The interpretation of this utterance by the agent depends not only on the agent's intentional state, but also crucially on its attentional state (focus stack). The details of Collagen's discourse interpretation algorithm are beyond the scope of this paper (see (Lesh, Rich, & Sidner 2001)), but the basic idea is to again apply Sidner's belief semantics, just reversing the speaker and hearer. Thus, after hearing the user's utterance, the agent adds the following belief:

Believe(agent, IntendThat(user, A({agent,user})))

If the agent's plan tree already contains a non-executed node for A, i.e., the agent already intended A, (and that part of the tree is on the focus stack) then the interpretation of the user's ProposeShould does not lead to any new intentions (plan nodes).

The more interesting case, however, is when A does not match an existing agent intention, but can contribute directly to the current discourse segment purpose (see (Grosz & Sidner 1986)). In this case, the result of the user's ProposeShould utterance is to add a subplan for A to the plan node for the current discourse segment purpose, as shown in Figure 5, where Y is assumed to be the current discourse segment purpose. [7] Currently (due to the agent compliancy discussed above), the belief annotation on this new node immediately gives it the full mutual belief semantics for A written above, which is not quite correct. More correctly, the agent may assume mutual belief only after it accepts the user's proposal.

To summarize, what we have seen here are two different communication routes to reach a Partial SharedPlan: either the agent communicates its intention to the user (and the user accepts), or the user communicates its intention to the agent (and the agent accepts).

Furthermore, in Collagen, plan recognition is applied to action proposals just like actions. What this means is that if the recipe library for the interpretation of the utterance in Figure 5 includes S and R, and these are the only recipes for Y and A respectively,[8] then the agent's plan tree after interpretation of the utterance will be exactly as shown in Figure 4 (except that B will not be marked as already executed).

Conclusion

In conclusion, although there is much further work to be done, we believe Collagen demonstrates that a single plan representation can support the fine-grained interleaving of plan generation, plan recognition and communication required for human-computer collaboration. Furthermore, we fervently believe that we would not have been able to get this far without the foundations provided by collaborative discourse theory.

References

Ferguson, G., and Allen, J. 2007. Mixed-initiative systems for collaborative problem solving. *AI Magazine* 28(2). Special Issue on Mixed-Initiative Assistants.

Grosz, B. J., and Kraus, S. 1993. Collaborative plans for group activities. In *Proc. 13th Int. Joint Conf. Artificial Intelligence*, 367–373.

Grosz, B. J., and Kraus, S. 1996. Collaborative plans for complex group action. *Artificial Intelligence* 86(2):269–357.

Grosz, B. J., and Sidner, C. L. 1986. Attention, intentions, and the structure of discourse. *Computational Linguistics* 12(3):175–204.

Grosz, B. J., and Sidner, C. L. 1990. Plans for discourse. In Cohen, P. R.; Morgan, J. L.; and Pollack, M. E., eds., *Intentions and Communication*. Cambridge, MA: MIT Press. 417–444.

Kumar, S., and Cohen, P. 2004. STAPLE: An agent programming language based on the joint intention theory. In *Proc. 3st Int. J. Conf. on Autonomous Agents and Multiagent Systems*, 1390–1391.

Lesh, N.; Rich, C.; and Sidner, C. 1999. Using plan recognition in human-computer collaboration. In *Proc. 7th Int. Conf. on User Modelling*, 23–32.

Lesh, N.; Rich, C.; and Sidner, C. 2001. Collaborating with focused and unfocused users under imperfect communication. In *Proc. 9th Int. Conf. on User Modelling*, 64–73.

Lochbaum, K. E. 1998. A collaborative planning model of intentional structure. *Computational Linguistics* 24(4):525–572.

Pollack, M. E. 1990. Plans as complex mental attitudes. In Cohen, P. R.; Morgan, J. L.; and Pollack, M. E., eds., *Intentions and Communication*. Cambridge, MA: MIT Press. chapter 5, 77–103.

Rich, C., and Sidner, C. 2007. DiamondHelp: A generic collaborative task guidance system. *AI Magazine* 28(2). Special Issue on Mixed-Initiative Assistants.

Rich, C.; Lesh, N.; Rickel, J.; and A., G. 2002. A plug-in architecture for generating collaborative agent responses. In *Proc. 1st Int. J. Conf. on Autonomous Agents and Multiagent Systems*.

Rich, C.; Sidner, C.; and Lesh, N. 2001. Collagen: Applying collaborative discourse theory to human-computer interaction. *AI Magazine* 22(4):15–25. Special Issue on Intelligent User Interfaces.

Rickel, J.; Lesh, N.; Rich, C.; Sidner, C.; and Gertner, A. 2002. Collaborative discourse theory as a foundation for tutorial dialogue. In *6th Int. Conf. on Intelligent Tutoring Systems*, 542–551.

Sadek, M.; Bretier, B.; and Panaget, F. 1997. Artimis: Natural dialogue meets rational agency. In *Proc. 15th Int. Joint Conf. Artificial Intelligence*, 1030–1035.

[7] If A cannot contribute to the current discourse segment purpose, then a new interruption segment will be created with A as the purpose. The full details of discourse interpretation are beyond the scope of this paper.

[8] If there is more than one recipe for Y, then the plan recognition is ambiguous and is deferred. If there is more than one recipe for A, then this choice is also deferred.

Sidner, C. L.; Lee, C.; Kidd, C.; Lesh, N.; and Rich, C. 2005. Explorations in engagement for humans and robots. *Artificial Intelligence* 166(1-2):104–164.

Sidner, C. L. 1994. An artificial discourse language for collaborative negotiation. In *Proc. 12th National Conf. on Artificial Intelligence*, 814–819.