Collagen™: Middleware for Building Mixed-Initiative Problem Solving Assistants

Charles Rich and **Candace L. Sidner**
Mitsubishi Electric Research Laboratories
201 Broadway
Cambridge, MA 02139
rich@merl.com

Abstract

Collagen™ is Java middleware for building mixed-initiative problem solving assistants, based on Grosz and Sidner's SharedPlan theory of collaborative discourse. The implementation includes a discourse state representation, comprised of a focus stack and a plan tree, as well as algorithms for discourse interpretation (including plan recognition) and discourse generation. Collagen has been used to build over a dozen research prototype systems.

Introduction

The concept of a mixed-initiative problem solving assistant is extremely closely related to the concepts of collaboration and collaborative discourse. *Collaboration* is a process in which two or more participants coordinate their actions toward achieving shared goals. Most collaboration between humans involves communication. *Discourse* is a technical term for an extended communication between two or more participants in a shared context, such as a collaboration. Collaborative discourse theory (see next section) thus refers to a body of empirical and computational research about how people collaborate. Essentially, what we have done in this project is apply a theory of human-human interaction to human-computer interaction.

In particular, we have taken the approach of adding a software agent (see Figure 1) to a conventional direct-manipulation graphical user interface. The name of our middleware, Collagen (for *Coll*aborative *agen*t), derives from this approach.[1] This approach mimics the relationships that typically hold when two humans collaborate on a task involving a shared artifact, such as two mechanics working on a car engine together or two computer users working on a spreadsheet together.

Notice that the software agent in Figure 1 is able both to communicate with and observe the actions of the user and vice versa. Among other things, collaboration requires knowing when a particular action has been done. In Collagen, this can occur two ways: either by a reporting communication ("I have done *x*") or by direct observation. Another symmetrical aspect of the figure is that both the user and the agent can interact with the application program.

For other overview articles on Collagen, see (Rich, Sidner, & Lesh 2001) and (Rich & Sidner 1998).

[1] Collagen is also a fibrous protein that is the chief constituent of connective tissue in vertebrates.

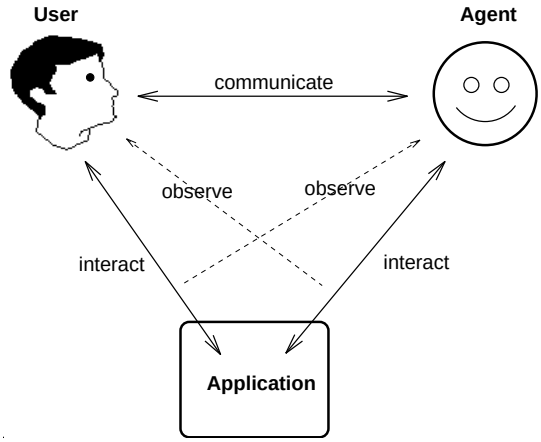


Figure 1: Setting for mixed-initiative problem solving.

Synopsis of Collaborative Discourse Theory

Grosz and Sidner (1986) proposed a tripartite framework for modelling task-oriented discourse structure. The first (*intentional*) component records the beliefs and intentions of the discourse participants regarding the tasks and subtasks ("purposes") to be performed. The second (*attentional*) component captures the changing focus of attention in a discourse using a stack of "focus spaces" organized around the discourse purposes. As a discourse progresses, focus spaces are pushed onto and popped off of this stack. The third (*linguistic*) component consists of the contiguous sequences of utterances, called "segments," which contribute to a particular purpose.

Grosz and Sidner (1990) extended this basic framework with the introduction of SharedPlans, which are a formalization of the collaborative aspects of a conversation. The SharedPlan formalism models how intentions and mutual beliefs about shared goals accumulate during a collaboration. Grosz and Kraus (1996) provided a comprehensive axiomatization of SharedPlans, including extending it to groups of collaborators.

Most recently, Lochbaum (1998) developed an algorithm for discourse interpretation using SharedPlans and the tripartite model of discourse. This algorithm predicts how conversants follow the flow of a conversation based on their understanding of each other's intentions and beliefs.

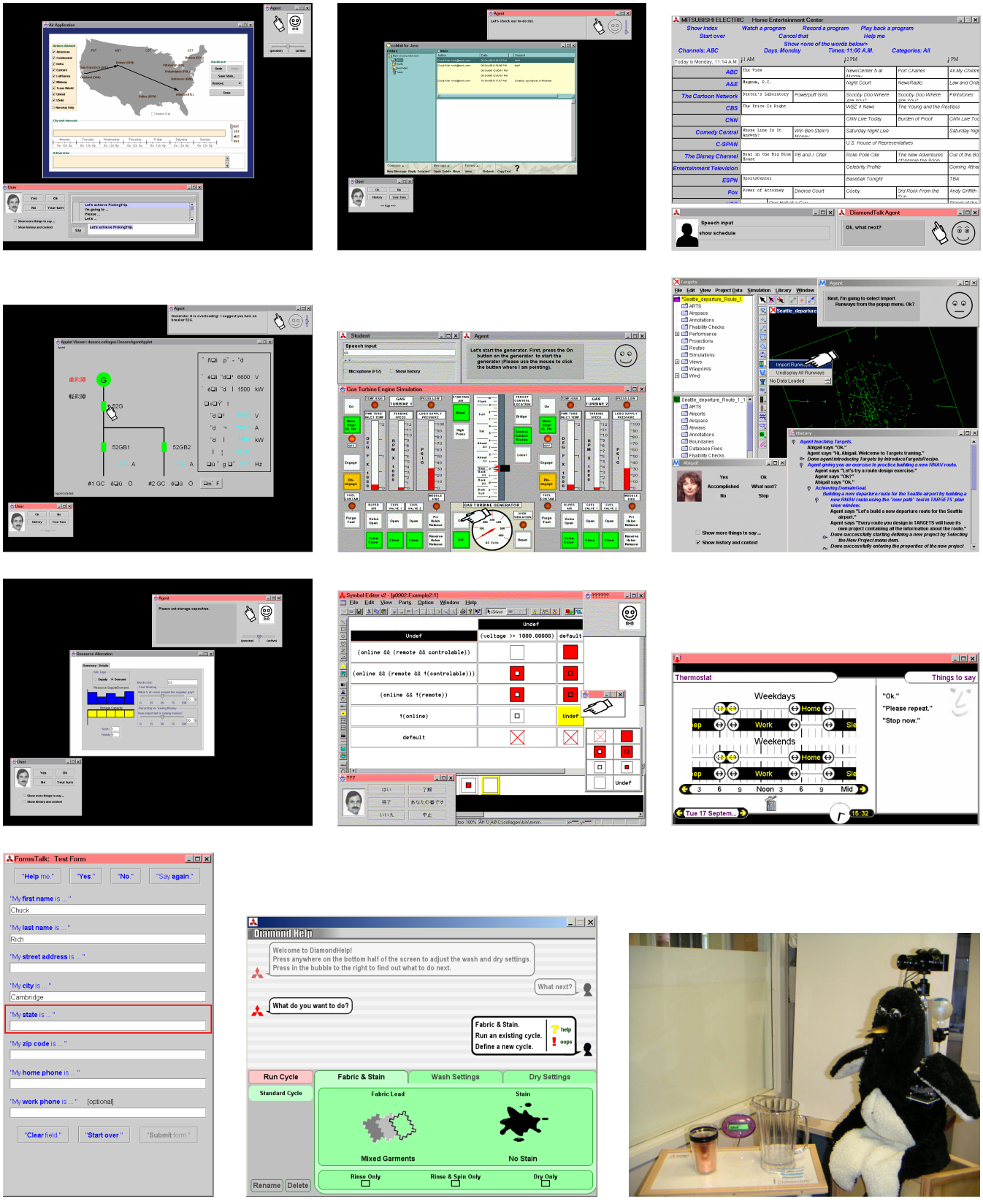
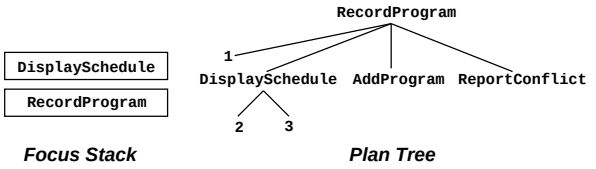


Figure 2: Screen shots of systems built with Collagen middleware (see text for descriptions).



Scheduling a program to be recorded.
1 User says "I want to record a program."
Done successfully displaying the recording schedule.
2 Agent displays recording schedule.
3 Agent says "Here's the schedule."
Next expecting to add a program to the recording schedule.
Expecting optionally to say there is a conflict.

Figure 3: Example discourse state and segmented interaction history for VCR assistant.

Examples

The true test of any middleware is how many times it has been reused. Figure 2 shows some of the more than a dozen systems that have been built using Collagen (from left to right, top to bottom):

- air travel planning assistant (Rich & Sidner 1998)
- email assistant (Gruen *et al.* 1999)
- VCR programming assistant (Sidner & Forlines 2002)
- power system operation assistant (Rickel *et al.* 2001)
- gas turbine engine operation tutor (Davies *et al.* 2001)
- flight path planning assistant (Cheikes & Gertner 2001)
- recycling resource allocation assistant
- software design tool assistant
- programmable thermostat helper (DeKoven *et al.* 2001)
- mixed-initiative multi-modal form filling
- intelligent help for programmable washer-dryer (Rich *et al.* 2005)
- robot hosting system (Sidner *et al.* 2005)

These systems range from small exercises to mature research prototypes; several of them have been developed outside of our laboratory. Communication between the user and the system has been variously implemented using speech recognition and generation, text, and menus (in both English and Japanese).

Discourse State

Participants in a collaboration derive benefit by pooling their talents and resources to achieve common goals. However, collaboration also has its costs. When people collaborate, they must usually communicate and expend mental effort to ensure that their actions are coordinated. In particular, each participant must maintain some sort of mental model of the status of the collaborative tasks and the conversation about them—we call this model the *discourse state*.

Among other things, the discourse state tracks the beliefs and intentions of all the participants in a collaboration and provides a focus of attention mechanism for tracking shifts in the task and conversational context. All of this information is used by an individual to help understand how the actions and utterances of the other participants contribute to the common goals.

In order to turn a computer agent into a collaborator, we needed a formal representation of discourse state and an algorithm for updating it. The discourse state representation currently used in Collagen, illustrated in Figure 3, is a partial implementation of Grosz and Sidner's SharedPlan theory; the update algorithm is described later in this section.

Collagen's discourse state consists of a stack of goals, called the *focus stack* (which will soon become a stack of focus spaces to better correspond with the theory), and a *plan tree* for each goal on the stack. The top goal on the focus stack is the "current purpose" of the discourse. A plan tree in Collagen is an (incomplete) encoding of a partial SharedPlan between the user and the agent. For example, Figure 3 shows the focus stack and plan tree immediately following the discourse events numbered 1–3 on the right side of the figure.

Segmented Interaction History

The annotated, indented execution trace on the right side of Figure 3, called a *segmented interaction history*, is a compact, textual representation of the past, present and future states of the discourse. We originally developed this representation to help us debug agents and Collagen itself, but we have also experimented with using it to help users visualize what is going in a collaboration (see discussion of "history-based transformations" in (Rich & Sidner 1998)).

The numbered lines in a segmented interaction history are simply a log of the agent's and user's utterances and primitive actions. The italic lines and indentation reflect Collagen's interpretation of these events. Specifically, each level of indentation defines a segment (see theory synopsis) whose purpose is specified by the italicized line that precedes it. For example, the purpose of the toplevel segment in Figure 3 is *scheduling a program to be recorded.*

Unachieved purposes that are currently on the focus stack are annotated using the present tense, such as *scheduling*, whereas completed purposes use the past tense, such as *done*. (Note in Figure 3 that a goal is not popped off the stack as soon as it is completed, because it may continue to be the topic of conversation, for example, to discuss whether it was successful.)

Finally, the italic lines at the end of each segment, which include the keyword *expecting*, indicate the steps in the current plan for the segment's purpose which have not yet been executed. The steps which are "live" with respect to the plan's ordering constraints and preconditions have the added keyword *next*.

Discourse Interpretation

Collagen updates its discourse state after every utterance or primitive action by the user or agent using Lochbaum's discourse interpretation algorithm with extensions to include plan recognition (see next section) and unexpected focus shifts (Lesh, Rich, & Sidner 2001).

According to Lochbaum, each discourse event is explained as either: (i) starting a new segment whose purpose contributes to the current purpose (and thus pushing a new purpose on the focus stack), (ii) continuing the current segment by contributing to the current purpose, or (iii) com-

```
public recipe RecordRecipe
             achieves RecordProgram {
   step DisplaySchedule display;
   step AddProgram add;
   optional step ReportConflict report;
   constraints {
      display precedes add;
      add precedes report;
      add.program == achieves.program;
      report.program == achieves.program;
      report.conflict == add.conflict;
   }
}
```

Figure 4: Example recipe from VCR task model.

pleting the current purpose (and thus eventually popping the focus stack).

An utterance or action contributes to a purpose if it either: (i) directly achieves the purpose, (ii) is a step in a recipe for achieving the purpose, (iii) identifies the recipe to be used to achieve the purpose, (iv) identifies who should perform the purpose or a step in the recipe, or (v) identifies a parameter of the purpose or a step in the recipe. These last three conditions are what Lochbaum calls "knowledge preconditions."

A *recipe* is a goal-decomposition method (part of a task model). Collagen's recipe definition language supports partially ordered steps, parameters, constraints, pre- and post-conditions, and alternative goal decompositions. Figure 4 shows the recipe used in Figure 3 to decompose the non-primitive `RecordProgram` goal into primitive and non-primitive steps. Collagen task models are defined in an extension of the Java language which is automatically processed to create Java class definitions for recipes and act types.

Our implementation of the discourse interpretation algorithm above requires utterances to be represented in Sidner's (1994) artificial discourse language. For our speech-based agents, we have used standard natural language processing techniques to compute this representation from the user's spoken input. Our menu-based systems construct utterances in the artificial discourse language directly.

Plan Recognition

Plan recognition (Kautz & Allen 1986) is the process of inferring intentions from actions. Plan recognition has often been proposed for improving user interfaces or to facilitate intelligent help features. Typically, the computer watches "over the shoulder" of the user and jumps in with advice or assistance when it thinks it has enough information.

In contrast, our main motivation for adding plan recognition to Collagen was to reduce the amount of communication required to maintain a mutual understanding between the user and the agent of their shared plans in a collaborative setting (Lesh, Rich, & Sidner 1999). Without plan recognition, Collagen's discourse interpretation algorithm onerously required the user to announce each goal before performing a primitive action which contributed to it.

Although plan recognition is a well-known feature of human collaboration, it has proven difficult to incorporate into practical computer systems due to its inherent intractability

Scheduling a program to be recorded.
1 User says "I want to record a program."
Done successfully displaying the recording schedule.
2 Agent displays recording schedule.
3 Agent says "Here's the schedule."
4 User says "Ok."
Done identifying the program to be recorded.
5 Agent says "What is the program?"
6 User says "Record 'The X-Files'."
Next expecting to add a program to the recording schedule.
Expecting optionally to say there is a conflict.

Figure 5: Continuing the interaction in Figure 3.

in the general case. We exploit three properties of the collaborative setting in order to make our use of plan recognition tractable. The first property is the focus of attention, which limits the search required for possible plans.

The second property of collaboration we exploit is the interleaving of developing, communicating about and executing plans, which means that our plan recognizer typically operates only on partially elaborated hierarchical plans. Unlike the "classical" definition of plan recognition, which requires reasoning over complete and correct plans, our recognizer is only required to incrementally extend a given plan.

Third, it is quite natural in the context of a collaboration to ask for clarification, either because of inherent ambiguity, or simply because the computation required to understand an action is beyond a participant's abilities. We use clarification to ensure that the number of actions the plan recognizer must interpret will always be small.

Our algorithm also computes essentially the same recognition if the user does not actually perform an action, but only proposes it, as in, "Let's achieve G." Another important, but subtle, point is that Collagen applies plan recognition to both user and agent utterances and actions in order to correctly maintain a model of what is mutually believed.

Discourse Generation

To illustrate how Collagen's discourse state is used to generate as well as interpret discourse behavior, we briefly describe below how the VCR agent produces the underlined utterance on line 5 in Figure 5, which continues the interaction in Figure 3.

The discourse generation algorithm in Collagen is essentially the inverse of discourse interpretation. Based on the current discourse state, it produces a prioritized list, called the *agenda*, of (partially or totally specified) utterances and actions which would contribute to the current discourse purpose according to cases (i) through (v) above. For example, for the discourse state in Figure 3, the first item on the agenda is an utterance asking for the identity of the program parameter of the AddProgram step of the plan for RecordProgram.

In general, an agent may use any application-specific logic it wants to decide on its next action or utterance. In most cases, however, an agent can simply execute the first item on the agenda generated by Collagen, which is what the VCR agent does in this example. This utterance starts a new segment, which is then completed by the user's answer on line 6.

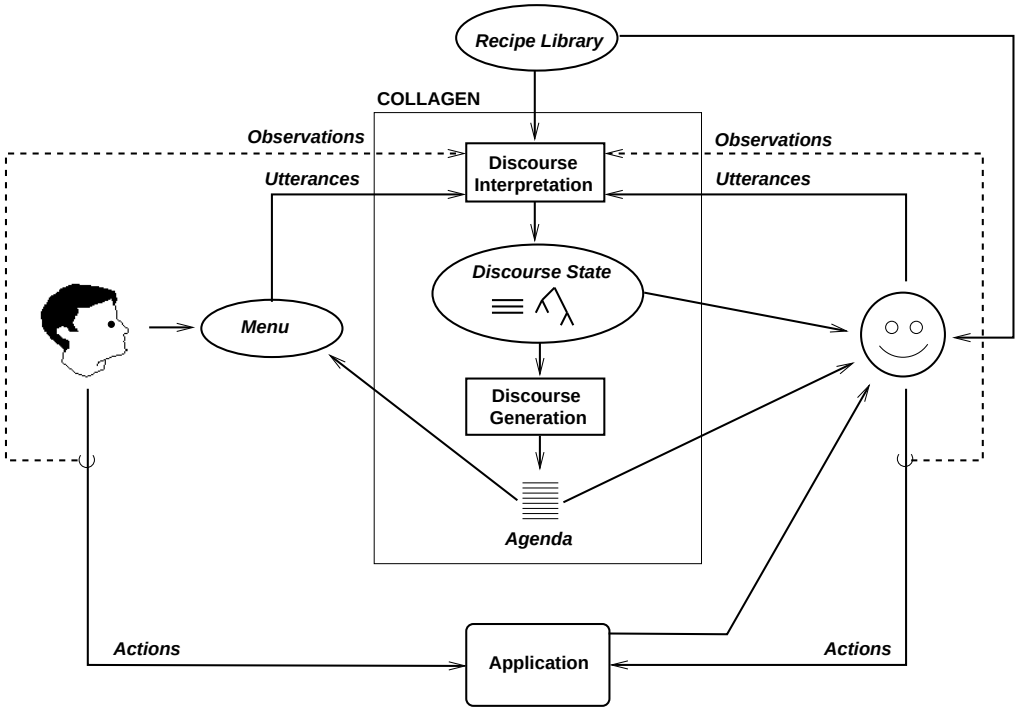


Figure 6: Architecture for mixed-initiative systems built with Collagen

System Architecture

Figure 6 shows how all the pieces described earlier fit together in the architecture of mixed-initiative problem solving assistant built with Collagen. This figure is essentially an expansion of Figure 1, showing how Collagen mediates the interaction between the user and the agent. Collagen is implemented using Java Beans™, which makes it easy to modify and extend this architecture.

The best way to understand the basic execution cycle in Figure 6 is to start with the arrival of an utterance or an observed action (from either the user or the agent) at the discourse interpretation module at the top center of the diagram. The discourse interpretation algorithm (including plan recognition) updates the discourse state as described above, which then causes a new agenda to be computed by the discourse generation module. In the simplest case, the agent responds by selecting and executing an entry in the new agenda (which may be either an utterance or an action), which provides new input to discourse interpretation.

In a system without natural language understanding, a subset of the agenda is also presented to the user in the form of a menu of customizable utterances. In effect, this is a way of using expectations generated by the collaborative context to replace natural language understanding. Because this is a mixed-initiative architecture, the user can, at any time, produce an utterance (e.g., by selecting from this menu) or perform an application action (e.g., by clicking on an icon), which provides new input to discourse interpretation.

In the simple story above, the only application-specific components an agent developer needs to provide are the recipe library and an API through which application actions can be performed and observed (for an application-independent approach to this API, see (Cheikes *et al.* 1999). Given these components, Collagen is a turnkey technology—default implementations are provided for all the other needed components and graphical interfaces, including a default agent which always selects the first item on the agenda.

In each of the four example applications in Figure 2, however, a small amount (e.g., several pages) of additional application-specific code was required in order to achieve the desired agent behavior. As the arrows incoming to the agent in Figure 6 indicate, this application-specific agent code typically queries the application and discourse states and (less often) the recipe library. An agent developer is free, of course, to employ arbitarily complex application-specific and generic techniques, such as a theorem proving, first-principles planning, etc., to determine the agent's response to a given situation.

Related Work

This work lies at the intersection of many threads of related research in artificial intelligence, computational linguistics, and user interface. We believe it is unique, however, in its combination of theoretical elements and implemented technology. Other theoretical models of collaboration (Levesque, Cohen, & Nunes 1990) do not integrate the intentional, attentional and linguistic aspects of collaborative discourse, as SharedPlan theory does. On the other hand, our incomplete implementation of SharedPlan theory in Collagen does not deal with the many significant issues in a collaborative system with more than two participants (Tambe 1997).

There has been much related work on implementing col-

laborative dialogues in the context of specific applications, based either on discourse planning techniques (Chu-Carroll & Carberry 1995; Ahn *et al.* 1994; Allen *et al.* 1996; Stein, Gulla, & Thiel 1999) or rational agency with principles of cooperation (Sadek & De Mori 1997). None of these research efforts, however, have produced software that is reusable to the same degree as Collagen. In terms of reuseability across domains, a notable exception is the Verbmobil project (Verbmobil 2000), which concentrates on linguistic issues in discourse processing, without an explicit model of collaboration.

Finally, a wide range of mixed-initiative interface agents (Maes 1994) continue to be developed, which have some linguistic and collaborative capabilities, without any general underlying theoretical foundation.

References

Ahn, R.; Bunt, H.; Benn, R.; Borghuis, T.; and Van Overveld, C. 1994. The DenK-architecture: A fundamental approach to user-interfaces. *Artificial Intelligence Review* 8:431–445.

Allen, J.; Miller, B.; Ringger, E.; and Sikorski, T. 1996. A robust system for natural spoken dialogue. In *Proc. 34th Annual Meeting of the Assoc. for Computational Linguistics*, 62–70.

Cheikes, B., and Gertner, A. 2001. Teaching to plan and planning to teach in an embedded training system. In *Proc. 10th Int. Conf. on Artificial Intelligence in Education*, 398–409.

Cheikes, B.; Geier, M.; Hyland, R.; Linton, F.; Riffe, A.; Rodi, L.; and Schaefer, H. 1999. Embedded training for complex information systems. *Int. J. of Artificial Intelligence in Education* 10:314–334.

Chu-Carroll, J., and Carberry, S. 1995. Response generation in collaborative negotiation. In *Proc. 33rd Annual Meeting of the Assoc. for Computational Linguistics*, 136–143.

Davies, J.; Lesh, N.; Rich, C.; Sidner, C.; Gertner, A.; and Rickel, J. 2001. Incorporating tutorial strategies into an intelligent assistant. In *Proc. Int. Conf. on Intelligent User Interfaces*, 53–56.

DeKoven, E.; Keyson, D.; and Freudenthal, A. 2001. Designing collaboration in consumer products. In *Proc. ACM Conf. on Computer Human Interaction, Extended Abstracts*, 195–196.

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.

Gruen, D.; Sidner, C.; Boettner, C.; and Rich, C. 1999. A collaborative assistant for email. In *Proc. ACM Conf. on Computer Human Interaction, Extended Abstracts*, 196–197.

Kautz, H. A., and Allen, J. F. 1986. Generalized plan recognition. In *Proc. 5th National Conf. on Artificial Intelligence*, 32–37.

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. Outstanding Paper Award.

Levesque, H. J.; Cohen, P. R.; and Nunes, J. H. T. 1990. On acting together. In *Proc. 8th National Conf. on Artificial Intelligence*, 94–99.

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

Maes, P. 1994. Agents that reduce work and information overload. *Comm. ACM* 37(17):30–40. Special Issue on Intelligent Agents.

Rich, C., and Sidner, C. 1998. Collagen: A collaboration manager for software interface agents. *User Modeling and User-Adapted Interaction* 8(3/4):315–350. Reprinted in S. Haller, S. McRoy and A. Kobsa, editors, *Computational Models of Mixed-Initiative Interaction*, Kluwer Academic, Norwell, MA, 1999, pp. 149–184.

Rich, C.; Sidner, C.; Lesh, N.; Garland, A.; Booth, S.; and Chimani, M. 2005. DiamondHelp: A new interaction design for networked home appliances. *Personal and Ubiquitous Computing*. To appear.

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. 2001. Using a model of collaborative dialogue to teach procedural tasks. In *Working Notes of AI-ED Workshop on Tutorial Dialogue Systems*, 1–12.

Sadek, D., and De Mori, R. 1997. Dialogue systems. In Mori, R. D., ed., *Spoken Dialogues with Computers*. Academic Press.

Sidner, C. L., and Forlines, C. 2002. Subset languages for conversing with collaborative interface agents. In *Int. Conf. on Spoken Language Processing*.

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.

Stein, A.; Gulla, J. A.; and Thiel, U. 1999. User-tailored planning of mixed initiative information seeking dialogues. *User Modeling and User-Adapted Interaction* 9(1-2):133–166.

Tambe, M. 1997. Towards flexible teamwork. *J. of Artificial Intelligence Research* 7:83–124.

Verbmobil. 2000. http://verbmobil.dfki.de.