

# Lessons Learned in Building Spoken Language Collaborative Interface Agents

Candace L. Sidner  
Carolyn Boettner  
Lotus Development Corporation  
Cambridge, MA 02142 USA  
csidner|carolyn\_boettner@lotus.com

Charles Rich  
Mitsubishi Electric Research Laboratory  
Cambridge, MA 02139 USA  
rich@merl.com

## Abstract

This paper reports on the development of two spoken language collaborative interface agents built with the Collagen system. It presents sample dialogues with the agents working with email applications and meeting planning applications, and discusses how these applications were created. It also discusses limitations and benefits of this approach.

## 1 Collaborative Agents

The underlying premise of the Collagen™ (for *Collaborative agent*) project is that software agents, when they interact with people, should be governed by the same principles that govern human-to-human collaboration. To determine the principles governing human collaboration, we have relied on research in computational linguistics on collaborative discourse, specifically within the SharedPlan framework of Grosz and Sidner (1986, 1990) (Grosz and Kraus, 1996, Lochbaum, 1998). This work has provided us with a computationally-specified theory that has been empirically validated across a range of

human tasks. We have implemented the algorithms and information structures of this theory in the form of a Java middleware component, a *collaboration manager* called Collagen, which software developers can use to implement a collaborative interface agent for any Java application.

In the *collaborative interface agent* paradigm, illustrated abstractly in Figure 1, a software agent is able to both communicate with and observe the actions of a user on a shared application interface, and vice versa. The software agent in this paradigm takes an active role in joint problem solving, including advising the user when he gets stuck, suggesting what to do next when he gets lost, and taking care of low-level details after a high-level decision is made.

The screenshot in Figure 2 shows how the collaborative interface agent paradigm is concretely realized on a user's display. The large window in the background is the shared application, in this case, the Lotus eSuite™ email program. The two smaller overlapping windows

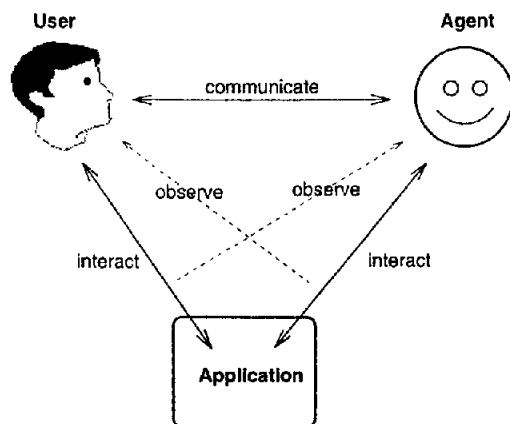


Figure 1: Collaborative interface agent paradigm.

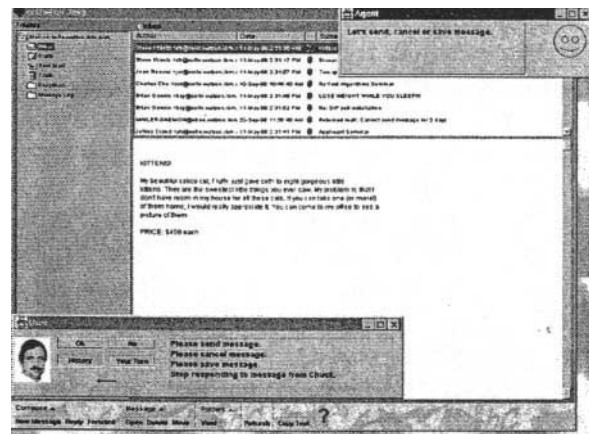


Figure 2: Interface for Collagen email agent.

in the corners of the screen are the agent's and user's *home windows*, through which they communicate with each other.

A key benefit of using Collagen to build an interface agent is that the collaboration manager automatically constructs a structured history of the user's and agent's activities. This *segmented interaction history* is hierarchically organized according to the goal structure of the application tasks. Among other things, this history can help re-orient the user when he gets confused or after an extended absence. It also supports high-level, task-oriented transformations, such as returning to an earlier goal. Figure 3 shows a sample segmented interaction history for the an email interaction.

To apply Collagen to a particular application, the application developer must provide an abstract model of the tasks for which the application software will be used. This knowledge is formalized in a *recipe library*, which is then automatically compiled for use by the interface agent. This approach also allows us to easily vary an agent's level of initiative from very passive to very active, using the same task model. For more details on the internal architecture of Collagen, see (Rich and Sidner, 1998).

We have developed prototype interface agents using Collagen for several applications, including air travel planning (Rich and Sidner, 1998), resource allocation, industrial control, and common PC desktop activities.

## 2 A Collaborative Email Agent

The email agent (Gruen et al., 1999) is the first Collagen-based agent we have built that supports spoken-language interaction. Our other agents avoided the need for natural language understanding by presenting the user with a dynamically-changing menu of expected utterances, which was generated from the current discourse state according to the predictions of the SharedPlan theory. Sample menus are displayed in Figure 2. The email agent, however, incorporates a speech and natural language understanding system developed by IBM Research, allowing users to collaborate either entirely in speech or with a mixture of speech and interface actions, such as selecting a message. More recently we have developed the Lotus Notes™ meeting planning agent, which

incorporates speech and sentence level understanding using the Java Speech API, as implemented by IBM. The JSAPI toolkit provides a parser, which we use with a vocabulary and grammar we developed for the domain of meeting planning. The tags produced by the Java Speech parser are interpreted with a set of semantic rules that produce internal structures used by the Collagen agent.

With the email application, the user can read, compose and send messages as one typically does with email. The Collagen email agent, called Daffy, performs actions requested by the user with speech and watches user interface actions. It can perform a few email actions on its own (such as opening and closing windows, and filling in the user's signature on email) and can also undertake actions that the user requests in spoken utterances. In the sample session shown in Figure 4, the agent keeps a todo list for the user, explains how to accomplish email tasks for a user who is new to email, answers user questions about what actions were taken in the interaction and offers suggestions about what to do next in the interaction as well as forming user requests.

To create the email agent, we built a recipe library about email, as required for the Collagen architecture, of about 55 actions and 32 recipes for doing those actions; the actions included GUI primitives such as sending a message, and high level actions such as reacting to a message.

## 3 A Collaborative Agent for Planning Meetings

Our experience with Daffy convinced us that collaborative agents would be more useful if they not only understood what the user was doing in the interface but could undertake more of the user's sub-goals and thus off-load some of the burden from the user. To explore this notion, we built Dotty, a Collagen agent, that works with a user who is planning a meeting with a customer, using Lotus Notes. As the dialogue in Figure 5 demonstrates, Dotty is able to take over many of the details of planning the meeting. Dotty uses a library that is far smaller than Daffy's: 19 actions and 5 recipes.

This dialogue begins with an overall goals of managing sales contacts and several sub-goals,

USER: "Your turn."  
 Done checking our to-do list.  
 AGENT: "Let's check our to-do list."  
 USER: "Ok."  
 AGENT: "The following items are on our to-do list."  
 AGENT: "Work on email."  
 AGENT: "Prepare a presentation about Java Beans."  
 AGENT: "That's all."  
 USER: "Ok."  
 Done working on email.  
 AGENT: "Let's work on email."  
 Done explaining how to work on email.  
 USER: "What should I do?"  
 AGENT: "First, you need a message."  
 USER: "Ok."  
 AGENT: "Then, you might store, print or delete the message."  
 Done reading message from Brian Gorwin <brg@dewey.watson.ibm.com> about WEEKLY SPEECH GROUP ....  
 Done explaining how to read a message.  
 Done agent selecting message from Brian Gorwin <brg@dewey.watson.ibm.com> about WEEKLY SPEECH GROUP ....  
 USER: "Select message from Brian Gorwin <brg@dewey.watson.ibm.com> about WEEKLY SPEECH GROUP ...."  
 AGENT: Select message from Brian Gorwin <brg@dewey.watson.ibm.com> about WEEKLY SPEECH GROUP ....  
 AGENT: Open message from Brian Gorwin <brg@dewey.watson.ibm.com> about WEEKLY SPEECH GROUP SEMINAR.  
 Done using the contents of message from Brian Gorwin <brg@dewey.watson.ibm.com> about WEEKLY SPEECH ....  
 AGENT: "Shall we use the contents of message from Brian Gorwin <brg@dewey.watson.ibm.com> about WEEKLY ...?"  
 Done scheduling meeting with Brian at three P.M. Thursday by scheduling.  
 Done agent setting up meeting with Brian.  
 USER: "Set up meeting with Brian."  
 AGENT: Set up meeting with Brian.  
 Done agent scheduling event tentatively at three P.M. Thursday.  
 AGENT: "Let's schedule event tentatively at a time."  
 USER: "Schedule event tentatively at three P.M. Thursday."  
 AGENT: Schedule event tentatively at three P.M. Thursday.  
 AGENT: "There is a conflict of meeting with Brian at three P.M. Thursday with meeting with Irene Landoz at three P.M. Thursday."  
 AGENT: "Let's remember to reschedule meeting with Irene Landoz at three P.M. Thursday."  
 USER: "Ok."  
 AGENT: Finalize meeting with Brian at three P.M. Thursday.  
 AGENT: Close message from Brian Gorwin <brg@dewey.watson.ibm.com> about WEEKLY SPEECH GROUP SEMINAR.  
 Done reacting to message from Brian Gorwin <brg@dewey.watson.ibm.com> about WEEKLY SPEECH ....  
 AGENT: "Shall we react to message from Brian Gorwin <brg@dewey.watson.ibm.com> about WEEKLY ...?"  
 Done agent forwarding message from Brian Gorwin <brg@dewey.watson.ibm.com> about WEEKLY ... to Steve Krantz.  
 USER: "Forward message from Brian Gorwin <brg@dewey.watson.ibm.com> about ... to Steve Krantz."  
 Done user saying identify the recipient as Steve Krantz.  
 AGENT: "Who is the recipient?"  
 USER: "The recipient is Steve Krantz."  
 AGENT: Forward message from Brian Gorwin <brg@dewey.watson.ibm.com> about ... to Steve Krantz.  
 Done filling in message from Candy Sidner <csidner@lotus.com> about Fwd:WEEKLY SPEECH GROUP SEMINAR.  
 AGENT: "Shall we fill in message from Candy Sidner <csidner@lotus.com> about Fwd:WEEKLY ...?"  
 USER: "Ok."  
 AGENT: Add text to message from Candy Sidner <csidner@lotus.com> about Fwd:WEEKLY ....  
 Stopped ccing message from Candy Sidner <csidner@lotus.com> about Fwd:WEEKLY SPEECH ....  
 AGENT: "Shall we cc message from Candy Sidner <csidner@lotus.com> about Fwd:WEEKLY SPEECH GROUP SEMINAR?"  
 USER: "No."  
 AGENT: "Ok, your turn."  
 Done sending, canceling or saving this message as a draft.  
 Done agent sending message from Candy Sidner <csidner@lotus.com> about Fwd:WEEKLY SPEECH ....  
 USER: "Send message from Candy Sidner <csidner@lotus.com> about Fwd:WEEKLY SPEECH GROUP ...."  
 AGENT: Send message from Candy Sidner <csidner@lotus.com> about Fwd:WEEKLY SPEECH ....  
 Done storing, printing or deleting this message.  
 AGENT: "Shall we store, print or delete this message?"  
 USER: "Did I forward message to Steve Krantz."  
 AGENT: "Yes, you did forward message from Brian Gorwin <brg@dewey.watson.ibm.com> about ... to Steve Krantz."  
 USER: "Done."  
 Done interruption agent checking history.  
 USER: "Please check history."  
 USER: "Done."

Figure 3: Sample segmented interaction history for an email interaction.



AGENT: "Let's manage sales contacts."  
USER: "Let's create profile for Miles White from Abbott Laboratories."  
AGENT: "What is the eemail address?"  
USER: "The emailAddress is mileswhite@abl.com."  
AGENT: "What is the workPhone?"  
USER: "The workPhone is 6173329280."  
AGENT: Goto profile for Abbott Laboratories.  
AGENT: Create profile for Miles White.  
AGENT: "Do you want to schedule a presentation for Miles White?"  
USER: "Schedule the meeting for December 9 from 2 to 4 pm."  
AGENT: Make meeting entry for Miles White on December 9.  
AGENT: "Is the meeting here or at the customer site?"  
USER: "The site is here."  
AGENT: "Which building?"  
USER: "The building is Rogers Street."  
AGENT: Locate room 5003 Center in Rogers Street.  
AGENT: "I'm going to reserve room 5003 Center in Rogers Street."  
USER: "Ok."  
AGENT: Reserve room 5003 Center in Rogers Street.  
AGENT: "Do you need any special equipment?"  
USER: "No."  
AGENT: "Okay."  
AGENT: "Who will attend the meeting?"  
USER: "Invite Fred Smith and Jerry Johnson."  
AGENT: Invite Fred Smith and Jerry Johnson.  
AGENT: "Will the customer need consulting services?"  
USER: "Ok."  
AGENT: "Identifying the consulting staff."  
USER: "Invite Pam Bridges from Lotus Consulting and Jimmy Barnes from IGS."  
AGENT: Invite Pam Bridges and Jimmy Barnes.  
AGENT: "The following people: Pam Bridges have a conflict with the meeting time.."  
USER: "Instead invite Julie Knox."  
AGENT: Invite Julie Knox instead of Pam Bridges.  
AGENT: "I will get back to you to confirm this meeting."  
USER: "Ok."  
AGENT: Send invitations to Fred Smith, Jerry Johnson, Julie Knox, and Jimmy Barnes.  
AGENT: "Do you want to see company information?"  
USER: "Show me information about Dover Hospital."  
AGENT: Goto profile for Dover Hospital.  
AGENT: "You need to plan a presentation for Miles White."  
AGENT: "Shall we find presentations about a subject?"  
USER: "No."  
AGENT: "Okay."  
AGENT: "I'm going to get back to you about meeting."  
AGENT: "The following people: Jimmy Barnes have a conflict with the meeting time.."  
AGENT: "This is message from Jimmy Barnes: I have to be in Cleveland on December 9. Jonathan Parks is available."  
USER: "Invite Jonathan Parks instead."  
AGENT: Invite Jonathan Parks instead of Jimmy Barnes.  
AGENT: "I will get back to you to confirm this meeting."

Figure 5: Sample session of Collagen agent using Lotus Notes for meeting planning (unsegmented).

including creating a profile for Miles White (which is displayed to the user as a Notes document), scheduling a meeting with Miles White (which the agent undertakes by itself using facilities in Lotus Notes), finding information about Dover Hospital (which is displayed as a Notes document), and a brief discussion about planning a presentation.

#### 4 Current Limitations

The spoken interaction of our two Collagen agents is limited by the range of utterances that the utterance understanding components can interpret. More significantly, we feel these agents are limited in dealing with spoken conversational errors, i.e. errors that arise either because the recognition system produces an error, or the semantic interpretation is faulty (even given the correct choice of words). Errors resulting from semantic mis-interpretation are especially important because often the content of the faulty interpretation is something that the agent can respond to and does, which results in the conversation going awry. In such cases we have in mind using the history based transformations possible in Collagen (c.f. (Rich and Sidner, 1998)) to allow the user to turn the conversation back to before where the error occurred.

Whether communicating by speech or menus, our agents are limited by their inability to negotiate with their human partner. For example, whenever one of our agents propose an action to perform that the user rejects (as in the email conversation in Figure 4, where the agent proposes filling in the cclist and the user says no), the agent currently does not have any strategies for responding in the conversation other than to accept the rejection and turn the conversation back to the user. We are in present exploring how to use a set of strategies for negotiation of activities and beliefs that we have identified from corpora of human-human collaborations. Using these strategies in the Collagen system will give interface agents a richer set of negotiation capabilities critical for collaboration.

Finally, our agents need a better model of conversational initiative. We have experimented in the Collagen system with three initiative modes, one dominated by the user, one by the agent and one that gives each some control

of the conversation. The dialogues presented in this paper are all from agent initiative. None of these modes is quite right. The user dominated mode is characterized by an agent that only acts when specifically directed to or when explicitly told to take a turn in the conversation, while the agent dominated mode has a very chatty agent that constantly offers next possible actions relevant to the collaboration. We are currently investigating additional modes of initiative.

The collaborative agent paradigm that we have implemented has several original features. The conversation and collaboration model is general and does not require tuning or the implementation of special dialogue steps for the agent to participate. The model tracks the interaction and treats both the utterances of both participants and the GUI level actions as communications for the discourse; it relates these to the actions and recipes for actions. The model has facilities for richer interpretation of discourse level phenomena, such as reference and anaphora, through the use of the focus stack. Finally, when we began this research, we were not certain that the Collagen system could be used to create agents that would interact with users for many different applications. Our experience with five different applications indicates that the model has the flexibility and richness to make human and computer collaboration possible in many circumstances.

#### References

- B. J. Grosz and S. Kraus. 1996. Collaborative plans for complex group action. *Artificial Intelligence*, 86(2):269-357, October.
- B. J. Grosz and C. L. Sidner. 1986. Attention, intentions, and the structure of discourse. *Computational Linguistics*, 12(3):175-204.
- B. J. Grosz and C. L. Sidner. 1990. Plans for discourse. In P. R. Cohen, J. L. Morgan, and M. E. Pollack, editors, *Intentions and Communication*, pages 417-444. MIT Press, Cambridge, MA.
- D. Gruen, C. Sidner, C. Boettner, and C. Rich. 1999. A collaborative assistant for email. In *Proc. ACM SIGCHI Conference on Human Factors in Computing Systems*, Austin, TX, May.
- K. E. Lochbaum. 1998. A collaborative planning model of intentional structure. *Computational Linguistics*, 24(4), December.
- C. Rich and C. Sidner. 1998. COLLAGEN: A collaboration manager for software interface agents. *User Modeling and User-Adapted Interaction*, 8(3/4):315-350.