Building a Bridge between Intelligent Tutoring and Collaborative Dialogue Systems

Jeff Rickel,1 Neal Lesh,2 Charles Rich,2 Candace L. Sidner2 and Abigail Gertner3

1 USC Information Sciences Institute, 4676 Admiralty Way, Marina del Rey, CA, 90292
rickel@isi.edu, http://www.isi.edu/isd/rickel

2 Mitsubishi Electric Research Laboratories, 201 Broadway, Cambridge, MA, 02139
lesh,rich,sidner@merl.com, http://www.merl.com/projects/collagen

3 MITRE Corporation, 202 Burlington Road, Bedford, MA, 01730
gertner@mitre.org, http://www.mitre.org/resources/centers/it/g068

1 Introduction

Our research objective is to develop computer tutors that collaborate with students on tasks in simulated environments. Towards this end, we seek to integrate two separate but related research threads: intelligent tutoring systems (ITS) and collaborative dialogue systems (CDS). Research on ITS [10] focuses on computer tutors that adapt to individual students based on the target knowledge the student is expected to learn and the presumed state of the student’s current knowledge. Research on CDS (e.g., [5]), with an equally long history, focuses on computational models of human dialogue for collaborative tasks.

Unfortunately, there has been a surprising lack of cross-fertilization between these two research areas. Work on tutorial dialogue for ITS has not leveraged general models of collaborative dialogue. Similarly, research on collaborative dialogue has focused on modeling conversations between peers or between an expert and novice, but has rarely addressed tutorial issues.

To help integrate ITS and CDS, we developed a tutorial agent in Collagen [6], a middleware system based on a long line of research on collaborative discourse. Collagen maintains a model of the discourse state shared by the user (e.g., student) and the computer agent (e.g., tutor). The discourse state includes information about the current focus of attention and the collaborators’ mutually believed plans. Agents constructed using Collagen use the discourse state to generate an agenda of candidate discourse acts, including both utterances and “physical” actions, and then choose one to perform or utter.

Our tutorial agent, Paco (Pedagogical Agent for Collagen), teaches students procedural tasks in simulated environments. While Paco can engage in slightly more sophisticated conversations than previous such tutors, our primary contribution is to show how a variety of tutorial behaviors can be expressed as rules for generating candidate discourse acts in Collagen. Translating behaviors developed in ITS into the framework of CDS is a first step towards building tutoring agents that can leverage advances in collaborative discourse theory. Also, since Paco is domain-independent, its tutorial actions can be added to the set of candidate discourse acts of any agent built with Collagen, allowing such agents to tutor in addition to their normal role as assistants.
2 Pedagogical Approach

Paco supports simulation-based training, in which students learn tasks by performing them in a simulation of the real work environment. Our pedagogical approach is based on the apprenticeship model of learning [1]. To implement this approach, Paco uses a student model to dynamically interleave demonstration and coached practice, using the approach introduced by Rickel [7].

Paco represents the procedures it will teach using Collagen’s declarative language for domain-specific procedural knowledge. This knowledge serves as a model of how domain tasks should be performed. Each task is associated with one or more recipes (i.e., procedures for performing the task), encoded in a relatively standard hierarchical plan representation.

3 Collagen as a Foundation for Teaching Procedural Tasks

Collagen’s main value for building tutoring systems is that it provides a general model of collaborative discourse based on well-established principles from computational linguistics. Previous tutoring systems for procedural tasks do not include dialogue managers with the same level of generality.

Based on the work of Grosz and Sidner [2], Collagen partitions the discourse state into three interrelated components: the linguistic structure, the attentional state, and the intentional structure. The linguistic structure groups the dialogue history into a hierarchy of discourse segments. Each segment is a contiguous sequence of actions and utterances that contribute to some purpose (e.g., performing a task or subtask). The attentional state, i.e., what the user and agent are talking about and/or working on now, is represented by a stack of discourse purposes called the focus stack [2]. The intentional structure represents the decisions that have been made as a result of all preceding actions and utterances. Collagen represents the intentional structure as plan trees, which are a partial implementation of SharedPlans [3].

Collagen’s discourse interpretation algorithm updates the discourse state given a new action or utterance by either the user or agent, and it uses plan recognition to recognize indirect relationships between an action or utterance and the current discourse state [4]. It has recently been extended to include “near-miss” plan recognition, which attempts to find plausible interpretations of student errors, providing a domain-independent capability for student diagnosis.

Collagen represents utterances using an artificial discourse language derived from earlier work by Sidner [9]. The language is intended to include the types of utterances that people use when collaborating on tasks. Currently, Collagen’s language includes utterance types for agreeing and disagreeing, proposing a task or action, indicating when a task has been accomplished, abandoning a task, asking about or proposing the value of a parameter to a task or action, asking or proposing how a task should be accomplished, and asking what should be done next. To bypass natural language understanding issues, Collagen provides a GUI through which the user constructs utterances. Optionally, Collagen can also use speech recognition software to allow the user to speak these utterances.

4 Tutorial Behaviors as Collaborative Discourse Acts

To produce Paco, we constructed a set of rules for generating candidate tutorial actions from the current discourse state, maintained by Collagen. We implemented rules for generating
a variety of tutorial actions including teaching steps of recipes, giving positive or negative feedback, ending interruptions by the student, and passing initiative to the student.

Paco uses several elements of the discourse state to generate its discourse acts including the focus of attention, the initiative, and plan trees. The focus of attention is used, for example, to avoid teaching a step unless its purpose is in focus. Collagen’s focus stack also indicates when the student has interrupted the current task, which causes Paco to generate a discourse act which would end the current interruption. Several of the tutorial actions are generated only on task steps that can be executed next based on precondition and ordering constraints; this information is also computed by Collagen during discourse interpretation. Additionally, Collagen’s near-miss recognition diagnoses incorrect actions which helps Paco give more constructive feedback.

The conditions for generating discourse acts represent necessary, but not sufficient, conditions for Paco to perform the act. An advantage of making explicit all necessary conditions for performing a discourse act is to make it easier to extend Paco with new discourse acts or extend other agents with the ability to perform Paco’s tutorial actions. Paco uses a simple priority scheme to choose which act to perform from the available candidates.

We are continuing to expand the range of Paco’s capabilities. For more technical details on our current version, see [8]. Paco thus far has been primarily a reimplementation (on a new foundation) of fairly standard ITS behaviors. As the next step, we plan to better leverage Collagen’s rich discourse state representation to implement aspects of tutorial dialogue that have not been treated in a fully general way in previous ITS work.

References


