Tutorial Strategies for Embedded Training ⋆

Abigail S. Gertner, Lisa A. Haverty, and Brant A. Cheikes[1]

The MITRE Corporation, Bedford, MA 01730

Complex information systems play a critical and growing role in today's workplace environment. Given the complexity of these systems, effective, high-quality training is increasingly being recognized as crucial. However, training is still consistently subject to budgetary pressures, and managers are demanding that employees be trained at or near the workplace in order to eliminate the cost of sending workers to remote training locations. As a result of these pressures, distance learning is proliferating, and computational training solutions are becoming increasingly attractive.

Our group is currently beginning a three year project in the area of *embedded training technology* – instructional tools embedded within the computing environment of the application(s) being used. Our goal is to develop an instructional agent to teach individual users how to operate complex software applications. The agent will adapt its instruction to the needs of the student by following explicit instructional strategies.

As a framework for the implementation our tutorial agent, we are using Collagen, an application-independent agent management platform developed by the Mitsubishi Electric Research Laboratory (MERL) [4]. Collagen provides tools for dialogue management which, when given a task-specific "recipe" library, will perform plan recognition, track the focus of attention of the tutor-student interaction, and maintain an "agenda" of actions that could complete the plan. Its underlying representation of goals and intentions is based on the SharedPlans formalism of Grosz and Kraus [2].

Collagen is primarily a tool for modelling the interaction between a user and an agent as they communicate with each other about activity in the domain application. It treats the agent as a black box which is capable of performing utterances and manipulating the application. The specific decision-making process guiding the agent's behavior is up to the developer of the agent. In this project, the agent will implement both a set of instructional strategies specifically designed to train users in the use of complex information applications, and a set of rules for determining the most appropriate strategy at any given time.

Our approach to developing instructional strategies for embedded training begins with the idea (first suggested by Norman [3]) that users of Graphical User Interfaces (GUIs) proceed through a cyclical process of forming intentions, selecting actions, acting, interpreting results, and modifying their intentions – in other words, they are engaged in dynamic planning. We believe that the "operator as dynamic planner" theory can help us develop effective instructional strategies for embedded training. Informal examples of potential alternative strategies include these:

⋆ This work is supported by the Mitre Technology Program, project #51MSR8AA-AA

- When the student is learning to perform a task, the agent might choose among three possible instructional strategies: (1) demonstrate the steps involved in the task, (2) walk the student through the task step by step, or (3) ask the student to perform the task.
- Students should be taught how to recognize, diagnose, and respond to execution failures. In the case of recognizing execution failures, the agent's strategy might be to immediately point to the part of the display that reflects the failure, or it might decide to let the student continue until she recognizes the failure on her own.

Our pedagogical agent will rely on a library of instructional strategies and principles drawn from the cognitive science literature. For example, the tutor design principles delineated in [1] provide some useful guidelines, such as facilitating successive approximations to the target skill, and minimizing working memory load. Decisions about how much support, or scaffolding, a student needs at any point during an exercise may be informed by research on identifying "teachable moments" and the student's zone of proximal development [5].

The appropriate choice of strategy will often depend on the student's level of domain knowledge, which is represented in the training system's student model. We are using a basic overlay student model which reflects the student's successful achievement of individual domain tasks. Additionally, the agent's strategy will depend on such considerations as whether it has given the same explanation before, how much time the student has spent on the current problem (and hence her potential level of frustration), and how many problems the student has solved correctly so far (she may need a boost to improve her motivation level).

Our initial work will focus on the implementation of one or two of the possible strategies listed above within a Collagen-aware agent, as well as the addition of a student model to the collagen framework. We will then apply these principles and our agent to a target application and examine the behavior and effectiveness of the agent in an embedded training environment. The dialogue management framework provided by Collagen will serve as a solid foundation for this work, allowing us to focus on the development and implementation of effective instructional strategies.

References

[1] J. Anderson, A. Corbett, K. Koedinger, and R. Pelletier. Cognitive tutors: Lessons learned. *The Journal of the Learning Sciences*, 4(2):167–207, 1995.

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

[3] D. A. Norman. *The Psychology of Everyday Things.* Basic Books, 1988.

[4] C. Rich and C. L. Sidner. COLLAGEN: A collaboration manager for software interface agents. *User Modeling and User-Adapted Interaction*, 8(3/4):315–350, 1998.

[5] L. S. Vygotsky. *Mind in Society: The Development of Higher Psychological Processes.* Harvard University Press, Cambridge, MA, 1978.