A Multi-Agent Architecture Implementation of Learning by Teaching Systems

Karun Viswanath, Bilikiss Adebiyi, Krittaya Leelawong, and Gautam Biswas
Dept. of EECS and ISIS
Vanderbilt University
Nashville, TN 37235
gautam.biswas@vanderbilt.edu

Abstract

Our group has been designing and implementing learning environments that promote deep understanding and transfer in complex domains. We have adopted the learning by teaching paradigm, and developed computer-based agents that students teach, and learn from this experience. The success of teachable agents has led us to develop a multi-agent architecture that will be used to develop extended instructional systems based on gaming environments.

1. Introduction

Advances in computer technology have facilitated the development of sophisticated computer-based Intelligent Tutoring Systems (ITS) [1]. The ITS paradigm is problem-based, with the system selecting problems for users to solve, and providing them feedback on their solutions. The tutoring paradigm suffers from its emphasis on localized feedback, and its inability to help students practice higher-order cognitive skills in complex domains, where problem solving requires active decision-making by learners in terms of setting learning goals and applying strategies for achieving these goals.

Studies of expertise [2] have shown that knowledge needs to be connected and organized around important concepts, and these structures should support transfer to other contexts. Other studies have established that improved learning happens when the students take control of their own learning, develop metacognitive strategies to assess what they know, and acquire more knowledge when needed. Thus the learning process must help students build new knowledge from existing knowledge (*constructivist learning*), guide students to discover learning opportunities while problem solving (*exploratory learning*), and help them to define learning goals and monitor their progress in achieving them (*metacognitive strategies*).

To advance the state of the art in the design and implementation of learning environments, we have adopted the learning by teaching paradigm. Teaching involves problem solving [3]. Learning-by-teaching is an open-ended and self-directed activity, which shares a number of characteristics with exploratory learning environments and constructivist learning. A natural goal for effective teaching is to gain a good understanding of domain knowledge before teaching it to others. Furthermore, teaching includes a process of structuring the knowledge in a form that it can be communicated to others. Studies have shown that teachers often reflect on their interactions with students during and after the teaching process [4]. Good learners bring structure to a domain by asking the right questions to develop a systematic flow for their reasoning. Good teachers build on the learners' knowledge to organize information, and in the process, they find new knowledge organizations, and better ways for interpreting and using these organizations in problem solving tasks.

From a system design and implementation viewpoint, this brings up an interesting question: *"How do we design learning environments based on the learning by teaching paradigm?*" This has led us to look more closely at the work on pedagogical and intelligent agents as a mechanism for modeling and analyzing student-teacher interaction. Section 2 summarizes some of the previous work in learning by teaching. Section 3 discusses Betty's Brain, our system for learning by teaching, which overcomes shortcomings of previous approaches. Section 4 describes an experimental study on fifth grade students that demonstrates the effectiveness of Betty's Brain. Section 5 extends the teachable agent implementation into a multi-agent architecture, which facilitates extended learning in gaming environments. Section 6 presents the conclusions of this paper.

2. Learning by Teaching: Previous Work

There have been a number of efforts to introduce intelligent agents into learning environments so as to create better and more human-like support for exploratory learning, and social environments that support tutor-tutee interactions and collaborative learning ([5],

[6]). Pedagogical agents are defined as “animated characters designed to operate in an educational setting for supporting and facilitating learning” [5]. They improve upon ITS schemes by adapting the agent’s behavior to the dynamic state of the learning environment, and can make the user aware of learning opportunities as they arise, much as a human mentor can. Agents extend the traditional text and graphics modalities of interaction with students using speech, animation, facial expressions and gestures, and this may help to improve student motivation and engagement. They may provide mechanisms for combining individualized and collaborative learning, by allowing multiple students and their agents to interact in a shared environment [7]. However, in most of these systems, the locus of control stays with the intelligent agent, which plays the role of the teacher or tutor.

There have been efforts to implement the learning by teaching paradigm using agents that learn from examples, advice, and explanations provided by the student-teacher (i.e., [9]). However, there has been no report on significant findings. Nichols [9] reported that students find it difficult to uncover, analyze, and learn from interactions with his system because the representations of the knowledge and the reasoning mechanisms are not visible to the user. Moreover, some of the systems provide outcome feedback or no feedback at all. It is well known that outcome feedback is less effective in supporting learning and problem solving than *cognitive feedback* [10].

On the positive side, students stated that they liked interacting with the agents that learn by teaching. Some studies also showed increased motivation and learning gains. But it was not clear from these studies that this approach helped achieve deep understanding and transfer in more complex domains. Therefore, we decided to adopt a new approach to designing learning by teaching environments that would support constructivist learning and the development of metacognitive strategies that helped students develop mechanisms for deep understanding and transfer. We also took on the challenge of teaching students who were novices in domain knowledge and teaching experience.

3. A New Approach: Betty’s Brain

Our Teachable Agents (TAs) are designed to simulate the behavior of a person’s thoughts about a domain. Learning empirical facts is important, but learning to employ these facts in organized problem structures is equally important. Therefore, we have structured TAs to simulate particular forms of thought that may help teacher-students structure their thinking about a domain. The particular agent that we discuss in this paper is Betty’s Brain, designed to teach middle school students about the concepts of interdependence and balance in river ecosystems [17, 18]. Betty’s Brain makes her qualitative reasoning visible through a dynamic, directed graph called a concept map [16].

Fig. 1 illustrates the interface of Betty’s Brain. Stu-

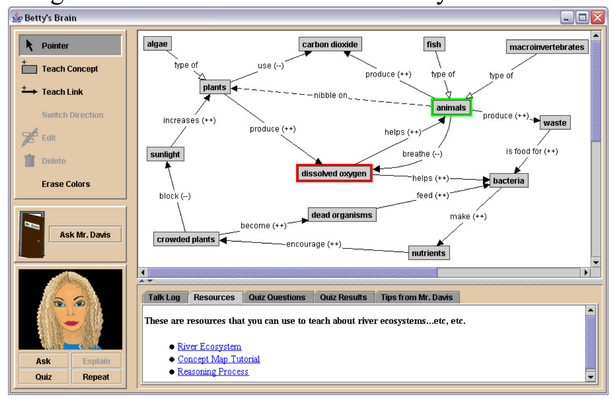

Figure 1: Betty’s Brain

dents explicitly teach Betty by creating a concept map (see top right pane of window) using a graphical drag and drop interface (top left of screen). Once taught, Betty can reason with her knowledge and answer questions. Users can formulate queries using the Ask button, and observe the effects of their teaching by analyzing Betty’s responses. Betty provides explanations for how she derives her answers by depicting the derivation process using multiple modalities: text, animation, and speech. The system uses qualitative reasoning to derive her answers to questions through a chain of causal inferences. Details of the reasoning and explanation mechanisms in Betty’s Brain are presented elsewhere [11].

The visual display of the face with animation in the lower left is one way in which the user interface attempts to provide engagement and motivation to users by increasing social interaction with the system. We should clarify that Betty does not use machine learning algorithms to achieve automated learning. Our focus is on the well-defined schemas associated with teaching that support a process of instruction, assessment, and remediation. These schemas help organize student interaction with the computer.

The system also includes sets of teacher-authorable quiz questions that enable students to get further feedback on Betty’s performance and, in this process, their teaching. A second agent, the Mentor agent informs Betty (and the student) if Betty’s answers are right or wrong. The mentor also provides directed hints to help the student debug errors in their concept map.

How do we deal with students who are novices in domain knowledge and in teaching? To accommodate this, the learning environment provides a number of scaffolds and feedback mechanisms that help students overcome their initial difficulties in learning and teaching about a complex domain. The scaffolds include well-organized online resources, structured quiz questions, and Mentor feedback that provides domain knowledge and information on how to teach and how to learn. We adopted the framework of *self-regulated learning,* where students are "*metacognitively, motivationally, and behaviorally participants in their own learning process*" [12]. Self-regulated learning strategies involve actions and processes that can help one to acquire knowledge and develop problem-solving skills [13]. Zimmerman describes a number of self-regulated learning skills that include goal setting and planning, seeking information, organizing and transforming, self-consequating, keeping records and monitoring, and self-evaluation.

Some of the self-regulated learning strategies (SRL) manifest through Betty's persona. They drive her interactions and dialog with the student, and make Betty more involved in the teach phase. For example, during concept map creation, Betty spontaneously tries to demonstrate *chains of reasoning*, and the conclusions she draws from this reasoning process. She may query the user, and sometimes remark (right or wrong) that an answer she is deriving does not seem to make sense. At other times, Betty will prompt the user to *formulate queries* to check if her reasoning with the concept map produces correct results. There are situations when Betty emphatically refuses to take a quiz because she feels that she has *not been taught enough*, or that the student has not given her *sufficient practice by asking queries* before making her take a quiz.

4. Experiments

A study was conducted on fifth graders to determine if Betty's Brain would be more effective in helping students to learn independently and gain deeper understanding of domain knowledge than an intelligent tutor designed as a pedagogical agent. To test this hypothesis, we divided up the fifth grade classroom into two groups using a stratified sampling method using standard achievement scores. One group used Betty's Brain (SRL system). They were told to teach Betty and help her pass a test so she could become a member of the school Science club. The second group used the pedagogical agent (ITS system). There was no Betty in their environment; all their interactions were with the Mentor agent, Mr. Davis. He asked students to create concept maps that could correctly answer the 16 quiz questions that were divided up into three groups. When the students' maps generated incorrect answers, Mr. Davis provided content feedback directly related to the quiz questions. On the other hand, the SRL mentor provided a variety of feedback, but only helped when the student asked questions. Both groups had access to the same set of domain resources and tools, but the learning direction of the SRL group was set to be internally guided while that of the ITS group externally-guided.

The students were separately introduced to their particular versions of the system, and then they worked for five 45-minute sessions over a period of three weeks to create their concept maps. Two delayed tests were conducted about seven weeks after the initial experiment: (i) a *memory test*, where students were asked to recreate their ecosystem concept maps from memory (there was no help or intervention when performing this task), and (ii) a preparation for future learning, a *transfer test*, where they were asked to construct a concept map and answer quiz questions about a new domain, the land-based nitrogen cycle with limited tools that included resources, the query mechanism, and the outcome feedback from the mentor agent.

A quick review of the data collected showed that there were notable differences in how the two groups used their systems during the initial learning phase. Fig. 2 shows the average number of resource, query, and quiz requests per session by the two groups. It is clear from the plots that the SRL group made a slow start, but after a couple of sessions this group showed a surge in map creation and map analysis activities, and their final concept maps and quiz performance were comparable to the ITS group. It seems the SRL group spent their first few sessions in learning self-regulation strategies (this can be attributed to Betty's behavior and the Mentor feedback), but once they learned them their performance improved significantly.

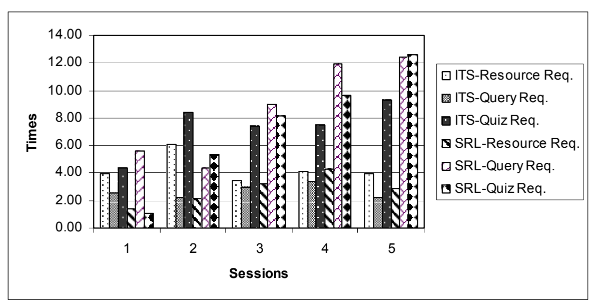

Figure 2. Resource, Query, and Quiz Requests

What about the impact of the system on long-term learning? Results of an ANOVA test on the memory test data with Tukey's LSD to make pair wise comparisons showed that the SRL group recalled significantly more causal links (3.3 to 2.0, $p < .05$) that were also in the expert concept map for the domain. Moreover, the effect of self-regulation strategies not only improved recall, but provide students with more skills for subsequent learning as well. The results from the transfer task indicate that the SRL group identified significantly more core concepts and causal links than the ITS group has (6.1 to 4.1 and 1.1 to 0.2 respectively, $p < .05$). The effects of teaching self-regulation strategies had a positive impact on the students' abilities to learn a new domain.

5. A Multi-agent Architecture for Betty's Brain

The SRL study results show that the effect of teaching and self-regulation may take longer to manifest, but their positive effects are more long lasting. This made us realize that the effectiveness of the system may be further improved, if students were presented with a series of challenges that built up on each other to keep students motivated and eager to learn for longer periods of time.

This led us to the idea of interactive gaming environments, with a structure where the game systematically introduces increasing levels of difficulty as students progress in their learning tasks. This framework could also support social interaction to promote collaborative learning among students. Students could first work on teaching their own agents, and after this agent reached a sufficient level of proficiency a group of agents could be brought together to solve more complex tasks. Alternately, competitive situations could be created among the students' agents, with the winning agents moving on to the next level of the game. To implement and evaluate this framework, we need to come up with a well-designed multi-agent architecture framework where agents can operate in different environments and interact with each other in a flexible, open-ended way.

5.1 Extending Betty's Brain

Our first step is to incorporate a sequence of challenges into Betty's Brain. Depending on how well she is taught, Betty will progress from one level to the next as she clears hurdles of increasing complexity. In this environment, students learn incrementally from a sequence of progressively complex challenges, relying on the scaffolding techniques that are gradually removed as students move to advanced states of the game.

To make Betty more mobile and adaptive, i.e., to enable her to work in different environments, we need a common message-passing and event handling architecture for communication between the agents and the agent and its environment. Also, encapsulating the agent components within standard interfaces will improve interoperability features. For example, this will enable a change in Betty's persona (e.g., allow Betty to use a different reasoning mechanism, or allow her to use a different set of emotions during her interactions) using replaceable plug-in modules, and the rest of the system would be unchanged.

5.2 New Architecture

The current multi-agent architecture in Betty's Brain uses four agents: the teachable agent, the mentor agent, and two auxiliary agents: the student agent and the environment agent. The last two agents help achieve greater flexibility by making it easier to move agents from one scenario to another without having to recode the communication protocols between agents. The student agent represents the interface of the student-teacher into the system. It provides facilities that allow the user to manipulate environmental functions and to communicate with the teachable agent. The environment agent is in essence a medium through which all of the agents (including the human users) communicate with each other and get to observe the global state of the system. This agent also has the ability to interpret new information, translate it into an appropriate form, and then direct it to the relevant agents. Having the environment modeled as an agent, makes it easier to swap environments, which is the equivalent of moving from one challenge problem to another.

5.3 Communication between Agents

All agents interact through the Environment Agent, which acts as a "Facilitator" [14]. The Environment Agent maintains information about the other agents and the services that they can provide. When an agent sends a request to the Environment Agent it: (i) forwards the request to an agent that can handle it, (ii) decomposes the request if different parts are to be handled by different agents and sends them to the respective agents, (iii) translates information between vocabularies to match an agent interface. This enables the environment to act as a mediator between all of the other agents.

During implementation of this system it is important to employ a standard communication language that agents will use to communicate with each other. Because of these needs, a variation of the FIPA ACL agent communication language [15] is used in the system. Each message sent by an agent contains a description of the message, message sender, recipient, recipient class and the actual content of the message. Communication is implemented using a Listener interface, where each agent listens only for messages from the Environment Agent and the Environment Agent listens for messages from all other agents.

The system is organized and implemented using a generic agent architecture shown in Fig. 3. Each agent has a Monitor, Decision Maker, Memory and an Executive. The Monitor listens for events from the environment, and using a pattern tracker converts them to an appropriate form needed by the agent. The decision maker, which contains two components: the *reasoner* and the *emotion generator,* is the agent's brain. It performs reasoning tasks (e.g., answering questions), executes local events, and determines the emotion state of the agent. The Executive posts multimedia (speech, text, graphics, animation) information from an agent to the environment. The executive also fires or broadcasts events in the environment. The Executive is made up of Agent Speech and Agent View, which handle speech and visual communication respectively.

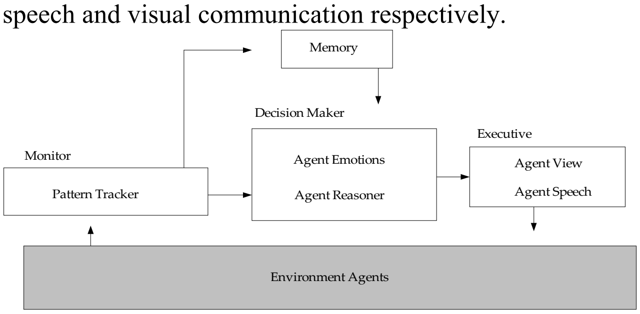

Figure 3: Agent Components

In the Teachable Agent for example, the monitor receives messages from the environment and patterns are stored in the pattern tracker. Memory records past events received from the monitor. The Decision Maker receives a request from the Monitor; within the Decision Maker, the Reasoner and Emotions use these requests along with memorized information to make a decision. A message is then sent to the executive who then decides on the modality with which to communicate this decision to the environment.

6. Conclusions

We have successfully demonstrated that our teachable agent environment provides mechanisms for deeper understanding and better transfer of domain knowledge. The next step has been the design and implementation of a multi-agent architecture to represent the learning environment. This has put us in a position where we can extend the capabilities of our system by incorporating them into gaming environments.

Acknowledgements

This work has been supported by a NSF ROLE Award #0231771. The assistance provided by Joan Davis, Kadira Belynne, Dan Schwartz, Nancy Vye, and John Bransford is gratefully acknowledged.

References

[1] Wenger, E., Artificial Intelligence and Tutoring Systems. 1987, Los Altos, California: Morgan Kaufmann Publishers.

[2] Glaser, R., Chi, M.T.H., and Farr, M.J. The Nature of Expertise. Erlbaum, Hillsdale, New Jersey, 1988.

[3] Artzt, A. F. and E. Armour-Thomas (1999). "Cognitive Model for Examining Teachers' Instructional Practice in Mathematics: A Guide for Facilitating Teacher Reflection." *Educational Studies in Mathematics* 40(3): 211-335.

[4] Chi, M. T. H., S. A. Siler, et al. (2001). "Learning from Human Tutoring," *Cognitive Science* 25(4): 471-533.

[5] G. Clarebout, J. Elen, W. L. Johnson, and E. Shaw (2002)."Animated Pedagogical Agents: An Opportunity to be Grasped?," *Journal of Educational Multimedia and Hypermedia*, 11: 267-286

[6] Johnson W., W. Rickel and Lester J. (2001)., "Animated Pedagogical Agents: Face-to-Face Interaction in Interactive Learning Environments", *International Journal of Artificial Intelligence in Education*11: 47-78

[7] Moreno, R. & Mayer, R. E. (2002). Learning science in virtual reality multimedia environments: Role of methods and media. *Journal of Educational Psychology*, 94: 598-610.

[8] Huffman, S. B. and J. E. Laird (1995). "Flexibly Instructable Agents." *Journal of Artificial Intelligence Research* 3: 271-324.

[9] Nichols, D. M. (1994). Intelligent Student Systems: an application of viewpoints to intelligent learning environments. *Unpublished Ph.D. thesis*, Lancaster University, Lancaster, UK.

[10] Butler, D. L. and P. H. Winne (1995). "Feedback and Self-Regulated Learning: A Theoretical Synthesis." *Review of Educational Research* 65(3): 245-281.

[11] Leelawong, K., Y. Wang, et al. (2001). Qualitative reasoning techniques to support learning by teaching: The Teachable Agents project. *Fifteenth Intl Workshop on Qualitative Reasoning*, San Antonio, Texas, AAAI Press.

[12] Zimmerman, B. J. (1989). "A Social Cognitive View of Self-Regulated Academic Learning." *Journal of Educational Psychology* 81(3): 329-339.

[13] Pintrich, P. R. and E. V. DeGroot (1990). "Motivational and self-regulated learning components of classroom academic performance." *Journal of Educational Psychology* 82(1): 33-40.

[14] Finin, T. and R. Fritzson (1994). "KQML as an Agent Communication Language" *3rd International Conference on Information and Knowledge Management (CIKM94)*, ACM Press

[15] Labrou. Y, T. Finin and Peng, Y. (1999) "Agent Communication Languages: The Current Landscape", *Intelligent Systems*, 14(2).

[16] Novak, J.D. (1996). Concept Mapping as a tool for improving science teaching and learning, in *Improving Teaching and Learning in Science and Mathematics*, D.F. Treagust, R. Duit, and B.J. Fraser, eds. Teachers College Press: London. 32-43.

[17] Biswas, G., Schwartz, D., Bransford, J., & the Teachable Agents Group at Vanderbilt (2001). Technology Support for Complex Problem Solving: From SAD Environments to AI. In Forbus & Feltovich (eds.), *Smart Machines in Education*, Menlo Park, CA: AAAI Press, 71-98.

[18] K. Leelawong, K., et al. (2003). *"Teachable Agents: Learning by Teaching Environments for Science Domains,"* Proc. 15th Innovative Applications of Artificial Intelligence Conf.*, Acapulco, Mexico, 109-116.*