

LEARNING BY TEACHING: A NEW AGENT PARADIGM FOR EDUCATIONAL SOFTWARE

Gautam Biswas and Krittaya Leelawong □ *Department of EECS and ISIS, Vanderbilt University, Nashville, TN, USA*

Daniel Schwartz □ *School of Education, Stanford University, Stanford, CA, USA*

Nancy Vye □ *Peabody School of Education, Vanderbilt University, Nashville, TN, USA*

The Teachable Agents Group at Vanderbilt □ *Vanderbilt University, Nashville, TN, USA**

□ *This paper discusses Betty's Brain, a teachable agent in the domain of river ecosystems that combines learning by teaching with self-regulation mentoring to promote deep learning and understanding. Two studies demonstrate the effectiveness of this system. The first study focused on components that define student-teacher interactions in the learning by teaching task. The second study examined the value of adding meta-cognitive strategies that governed Betty's behavior and self-regulation hints provided by a mentor agent. The study compared three versions: a system where the student was tutored by a pedagogical agent, a learning by teaching system, where students taught a baseline version of Betty, and received tutoring help from the mentor; and a learning by teaching system, where Betty was enhanced to include self-regulation strategies, and the mentor provided help on domain material on how to become better learners and better teachers. Results indicate that the addition of the self-regulated Betty and the self-regulation mentor better prepared students to learn new concepts later, even when they no longer had access to the SRL environment.*

The development of intelligent agent technologies bodes well for computer-based educational systems. From the perspective of interface design, what the desktop metaphor is to organizing one's documents, the agent metaphor is to education. People have a number of pre-existing social schemas

*TAG-V: Members of this group who contributed to this work include Joan Davis, Kadira Belyne, Karun Viswanath, John Bransford, and Thomas Katzlberger.

This work has been supported by a NSF ROLE Award #0231771. The assistance provided by the rest of the Teachable Agents Group and others, especially John Kilby, Bobby Bodenheimer, and Janet Truitt, is gratefully acknowledged.

Address correspondence to Gautam Biswas, Box 1824 Sta B, Department of EECS, Vanderbilt University, 400 24th Avenue South, Room 254, Featheringill Hall, Nashville, TN 37235, USA. E-mail: gautam.biswas@vanderbilt.edu

that they bring to agent systems and that help guide them without step-by-step prompting. These include the interpretation of gestures and other attention focusing behaviors (Rickel and Johnson 1998), affective attributions and responses (Reeves and Nass 1996) and the use of complex behavioral sequences like turn taking (Cassell 2004). From the instructional perspective, the agent metaphor can harvest the large repertoire of methods that people use to learn through social interaction. Intelligent tutoring systems (Wenger 1987) capitalized on social models of learning by mimicking some aspects of one-on-one human tutoring, though the original work was not based on the agent metaphor. When combined with an agent paradigm, intelligent tutors provide a wealth of possibilities; for example, an appropriately structured intelligent agent can ask questions and raise a user's sense of responsibility, and therefore, focus and motivate them to learn more deeply. Also, because the agent is a computer application, it can go beyond what human teachers can do. For example, besides infinite patience, a computer application can literally make an agent's thinking visible by providing a graphical trace of its reasoning (Schwartz et al. in press). The potential of agent technologies for education are great. In this paper, we describe our efforts at developing *teachable agents*. Teachable agents are computer agents that students teach, and in the process, learn themselves.

There are two key challenges to making headway with pedagogical agents. One challenge is the development of agent architectures and software tools that are well-suited to educational applications. For example, we recently began merging one of our agent applications with a three-dimensional gaming world. The gaming system provided primitives for bleeding and shooting, whereas we needed primitives for confidence and sharing. In this paper, one of our goals is to sketch an agent architecture that permits us to flexibly port and combine the capacities of our agents.

The second challenge is to determine effective ways to leverage the agent metaphor for educational purposes. While many researchers are attempting to make agents as realistic as possible, it is not clear that we want agents to imitate what currently exists in education. For example, a research review by Ponder and Kelly (1997) determined that the science education crisis in U.S. schools has been present for over four decades. Science curricula still needs to work on increasing student literacy, encouraging conceptual understanding, motivating students, and developing concrete problem solving skills (Ponder and Kelly 1997; Bransford et al. 2000). Unfortunately, current pedagogical practices tend to emphasize memorization, which provides students with limited opportunities and little motivation to develop "usable knowledge." Blindly imitating prevalent instructional interactions, like drill and practice, seems like a bad idea.

There have been significant developments in the area of pedagogical agents, which have been defined as *animated characters designed to operate*

in educational settings for supporting and facilitating learning (Shaw et al. 1999). These agents are designed to play the role of a coach (Burton and Brown 1982). When needed, they intervene to demonstrate and explain problem-solving tasks. Some systems use AI techniques to tailor their level of explanation and demonstration to the student's proficiency level. Multimodal delivery that combines text, speech, animation, and gestures has been used to improve communication between the agent and the student (Lester et al. 1997; Moreno et al. 2000). In some cases, there have been attempts to implement a two-way dialogue, using natural language understanding mechanisms. In almost all cases (the exception is a system called Carmen's Bright IDEAS: [Marsella et al. 2000]), the pedagogical agents follow a sequential, structured approach to teaching, and they significantly reduce opportunities for exploration and discovery—characteristics that are important for learning complex problem solving (Lajoie 1993; Crews et al. 1997).

In our work, we have attempted to support three critical aspects of effective interactions that the learning literature has identified as important. One form of interaction helps students *develop structured networks of knowledge* that have explanatory value. Not only do people remember better when information is connected in meaningful ways, they are also more able to apply it to new situations that are not identical to the original conditions of acquisition (Bransford et al. 1989).

A second form of interaction needs to help students *take responsibility* and make decisions about learning. Schwartz, Bransford, and Sears (in press) describe a conversation with school superintendents. These researchers explained that they studied learning and asked the superintendents what they wanted for their students. The superintendents' unanimous response was that they wanted their students to be able to learn and make decisions on their own once they left school. Thus, their goal for instruction was not to train students in everything they would need to know, but rather, it was to *prepare the students for future learning* (Bransford and Schwartz 1999). Instruction that spoon feeds students does not work as well for future learning as does instruction that helps students take on the responsibility of exploring and inventing their own solutions before they receive the canonical answer (Schwartz and Martin 2004).

The third aspect that has shown exceptional importance for learning is the development of *reflection or meta-cognitive skills* that include monitoring the quality of one's knowledge and learning decisions. These skills are ideally developed through social and modeling interactions (Palinscar and Brown 1984) and therefore, agent technologies are extremely well-suited to developing reflective meta-cognition (Shimoda et al. 2002).

In our work on teachable agent technologies, we have leveraged a particular model of social learning—learning by teaching—which we believe

does a good job of capturing each of these aspects of effective learning. We say much more about teachable agents next, including their motivation in the learning literature, their computational aspects, and the results of empirical studies on their effectiveness.

LEARNING BY TEACHING

The cognitive science and education literature supports the idea that teaching others is a powerful way to learn. Research in reciprocal teaching (e.g., Palinscar and Brown [1984]), peer-assisted tutoring (e.g., Willis and Crowder [1947]; Cohen *et al.* [1982]), programming (e.g., Dayer [1996]; Kafai and Harel [1991]; Papert [1993]), small-group interaction (e.g., Webb [1983]), and self-explanation (Chi *et al.* [1994]) hints at the potential of learning by teaching. Bargh and Schul (1980) found that people who prepared to teach others to take a quiz on a passage learned the passage better than those who prepared to take the quiz themselves. The literature on tutoring has shown that tutors benefit as much from tutoring as their tutees (Graessaer *et al.* 1995; Chi *et al.* 2001). Biswas and colleagues (2001) report that students preparing to teach made statements about how the responsibility to teach forced them to gain deeper understanding of the materials. Other students focused on the importance of having a clear conceptual organization of the materials. Beyond preparing to teach, actually teaching can tap into the three aforementioned critical aspects of learning interactions: structuring, taking responsibility, and reflecting.

For structuring, teachers provide explanations and demonstrations during teaching and receive questions and feedback from students. These activities can help teachers structure their own knowledge. For example, teachers' knowledge structures become better organized and differentiated through the process of communicating key ideas and relationships to students and reflecting on students' questions and feedback. Our studies have found that students who teach develop a deeper understanding of the domain, and can express their ideas better than those who study the same material and are asked to write a summary (Biswas *et al.* 2001). For taking responsibility, teaching is frequently an open-ended, self-directed problem-solving activity (Artz and Armour-Thomas 1999). Teachers need to take on the responsibility of deciding which content is most relevant. Additionally, there is a strong motivation component of teaching where the teacher needs to take responsibility (and joy) for the learning of their pupils. Finally, for reflection, effective teaching requires the explicit monitoring of how well ideas are understood and used. Studies have shown that tutors and teachers often reflect on their interactions with students during and after the teaching process (Chi *et al.* 2001). This reflection aids teachers

in evaluating their own understanding of domain concepts as well as the methods they have used to convey this understanding to students.

Each of these idealized benefits of teaching will largely depend on the context and resources for instruction, as well as the quality of the students one teaches. Most instructors can distinguish those students who push their own thinking from those who provide little intellectual stimulation. When reviewing the literature on computer based learning-by-teaching environments, there have been some hints of how to realize these positive benefits, though not all the work is equally relevant. For example, the intelligent tutoring system paradigm has typically led students through procedural sequences, and therefore, it is not ideal as a guide for how to design environments that help students develop responsibility and reflection.

One promising body of techniques comes from work on learning by induction, though in large part, this work has emphasized automated learning by the agent instead of focusing on explicit teaching by the user. This work includes learning from examples, advice, and explanations (e.g., Huffman and Laird [1995]; Srinivas et al. [1991]). In Huffman and Laird's system (1995), agents learn tasks through tutorial instructions in natural language. Users have some domain knowledge, which they refine by looking at the agents behaviors. Lieberman and Maulsby (1996) focus on teaching "*instructible agents*" by example and by providing advice. Agents learn by observing user actions, sometimes by being told what is relevant, and sometimes by identifying relevant information, applying it, and learning through the correction of mistakes. Michie et al. (1989) developed the Math Concept Learning System for solving linear equations. Users supplied the strategies for solving problems by entering example solution traces, and the system learned via an inductive machine learning algorithm, ID3 (Quinlan 1986). In comparison with other control conditions (an equation solving environment, a passive agent), students seemed to learn better with this agent. For this domain, machine learning techniques when coupled with learning by teaching systems proved to be useful in helping students to learn.

Research projects that emphasize learning by programming share family resemblances to learning by teaching in that students "inform" the computer to take specific actions. But these approaches have not fully capitalized on the agent metaphor and the host of social schema that it provides. Repenning and Sumner (1995) for example, developed visual programming environments that reduce the overhead of learning to program agents. Smith et al.'s (1997) Cocoa program (previously KidSim) allows young users to program their agents by example. Once created, they become alive in the environment and act according to their preprogrammed behavior. Other work, such as the Persona project (Ball et al. 1997), has focused on sophisticated user interactions, communication,

and social skills. Research that has specifically leveraged the teaching metaphor has shown positive results that may go beyond the benefits of programming. Obayashi *et al.*'s study (2000) reported significant learning gains in subjects using their learning-by-teaching system compared to a traditional computer-assisted instructor (CAI). Chan and Chou's (1997) study using reciprocal tutoring methods concluded that learning by teaching is better than studying alone. Hietala and Niemirepo (1998) designed the EduAgents environment that supports the solving of elementary equations. Their goal was to study the relation between the competency of agents designed as learning companions and student motivation. They found that students with higher cognitive ability preferred interacting with strong agents, i.e., agents that produced correct results and used a "knowing" manner of speaking. The weaker students preferred weak agents that initially made mistakes and were not confident of their answers, but improved their performance as the students improved in their abilities. A third group, introverted students, initially used the weak and strong agents equally, but as they progressed to more complex problems, they preferred the strong agents. All groups showed marginal improvements in their post-test scores. DENISE (Development for an Intelligent Student Environment in Economics) (Nichols 1994) used a simulated student to take on the role of a tutee in a peer-tutoring situation. The agent employed a Socratic teaching strategy to acquire knowledge from the student, and create a causal structure, but this structure was not made visible to the user. As a result, when the agent quizzed students to get them to self-assess the structures they had created, these students often failed to understand the agent. They could not remember what concepts they had previously taught, and found the interactions to be unnatural and frustrating.

In our design of teachable agents, which we describe next, we have tried to work around two aspects of prior systems that we think may limit the power of learning by teaching. One aspect is that these systems focus on learning from the activities of users during problem solving and the examples that they provide, but the representations of that knowledge and the reasoning mechanisms are not made explicit to the users. They are like many full-blown simulations and videogames, where the underlying logic of the program is hidden from the student rather than made transparent and more easily learned. Thus, students may find it difficult to uncover, analyze, and learn from the activities of these agents, and they are less likely to develop the level of structured knowledge that is embodied in the agent itself. The second aspect of these systems that may not do full justice to learning by teaching is that they have followed the tendency of the intelligent tutoring system paradigm to "over control" the learners' actions. In the typical session, the learners solve a set of small problems related to the curriculum that have structured answers in the order that the

software tutor selects. Other than local problem-solving decisions, the computer is in charge of the learning. We suspect (and test below) that this limits students' opportunities to develop responsibility and skills for learning on their own.

A New Approach to Learning by Teaching

Unlike previous studies on learning by teaching, the current took on the challenge of teaching students who were novices in the domain, and also had very little experience in teaching others. We have designed teachable agents (TAs) to provide important structures that help shape teacher thinking (Biswas et al. 2001;2004). Each agent manifests a visual structure that organizes specific forms of knowledge organization and inference. In general, our agents try to embody four principles of design:

- Teach through visual representations that organize the reasoning structures of the domain (e.g., directed graphs, matrices, etc.).
- Build on well-known teaching interactions to organize student activity (e.g., teaching by "laying out," teaching by example, teaching by telling, teaching by modeling).
- Ensure the agents have independent performances that provide feedback on how well they have been taught (each agent depends on a distinct AI reasoning technique: qualitative reasoning, logic, and genetic algorithms).
- Keep the start-up costs of teaching the agents very low (compared to programming). This occurs by only implementing one model of reasoning, rather than attempting to provide a complete system with multiple representations and reasoning elements.

We have designed a number of agents that aid students in a variety of domains: mathematics, science, and logical reasoning (Biswas et al. 2004; Leelawong et al. 2003; Leelawong et al. 2002; Biswas et al. 2001; Schwartz et al., to appear). One of our agents, Betty, described next, makes her qualitative reasoning visible through a dynamic, directed graph called a concept map. The fact that TAs represent knowledge structures rather than the referent domain is a departure from many simulations. Simulations often show the behavior of a physical system, for example, how an algal bloom increases the death of fish. TAs, however, simulate the behavior of a person's thoughts about a system. This is important because the goal of learning is often to simulate an expert's reasoning processes about a domain, not the domain itself. Learning empirical facts is important, but learning to think with the expert problem solving theory that organizes those facts is equally important. Therefore, we have structured the agents to simulate

particular forms of thought that may help teacher-students structure their thinking about a domain.

BETTY'S BRAIN: A LEARNING-BY-TEACHING ENVIRONMENT (VERSION 1)

The Betty's Brain system, discussed in this paper, is designed to teach middle school students about interdependence and balance among entities in a river ecosystem. A primary consideration in the design was to ensure that it could be used by students who had almost no experience in teaching, and little prior knowledge of the domain. Therefore, we designed the computer-based agent to have stable and predictable characteristics, for example, Betty never forgets what she is taught. Also, unlike other intelligent agents, Betty does not use machine learning techniques to learn from examples or by induction; she knows and reasons with what she has been taught by the student. This made it easier for our student teachers to understand the agent's thinking and reasoning processes. The use of a computer agent also absolved us of the responsibility of "*bad teachers frustrating students*," because there is little the student could do to upset the psyche or destroy the motivation of the software agent. On the other hand, in preliminary work with children, we found that students were very motivated by the agents, and they readily took on the responsibility for their learning, and they actively worked to make structured knowledge.

In the next sections, we describe the basic agent architecture of one agent, Betty's Brain. We follow with an example of research conducted on this basic system. We then describe how we improved the system to help young students become more effective teachers (and students) by providing metacognitive support. One of the exciting aspects of this improved system is that it had consequences for how well students were able to learn later, even when they were not using the system anymore.

Agent Architecture for Betty's Brain

A learning by teaching system requires a shared representation and reasoning structure that the student teacher shares with the teachable agent. For Betty, the representational structure had to be intuitive and easily understandable by fifth grade students, and at the same time sufficiently expressive to help these students create and organize, and reason with the knowledge structures to solve problems. A widely accepted technique for constructing knowledge is the concept map (Novak 1996). Concept maps provide a mechanism for structuring and organizing knowledge into hierarchies, and allow the analysis of phenomena in the form of cause-effect relations (Kinchin and Hay 2000; Stoyanov and Kommers 1999). This makes them amenable to

Figure 1 illustrates the interface of Betty's Brain. Students use a graphical point and click interface, in particular the *teach concept*, *teach link*, and *edit* buttons to create and modify their concept maps in the top pane of the window. 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 answers. When asked (by clicking the *explain* button), Betty provides explanations for her answers by depicting the derivation process using multiple modalities: text, animation, and speech. Betty uses qualitative reasoning to derive her answers to questions through a chain of causal inferences. The visible face in the lower left, which animates as it speaks, is one way in which the user interface attempts to provide engagement by increasing the social interaction between Betty and the student. This should help the student's motivation to learn. (Reeves and Nass 1996).

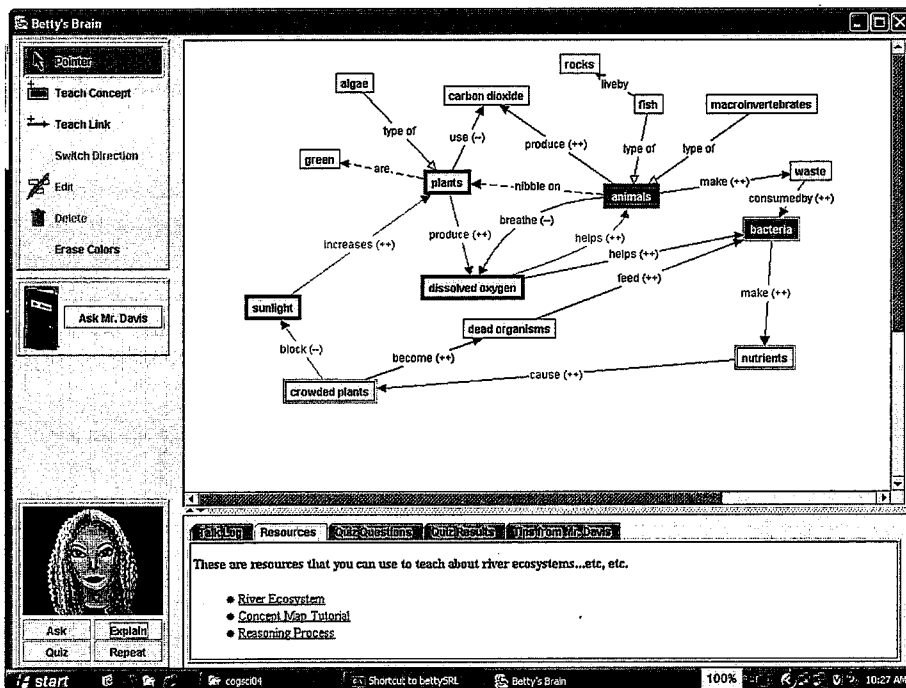


FIGURE 1 Betty's Brain: Interface.

The system implementation adopts a generic agent architecture illustrated in Figure 2. It includes components of a traditional intelligent architecture, e.g., Arkin's Autonomous Robot Architecture (AuRA) that includes five primary components: (i) a perception system, (ii) a knowledge base for maintaining a priori and acquired world knowledge, (iii) a planning mechanism (hierarchical deliberative plus a reactive planner), (iv) a motor subsystem that is the interface to the physical robot control mechanisms, and (v) a homeostatic control system that monitors the internal state of the system (Arkin 1991). Betty's Brain uses a simplified version of this standard agent architecture.

The *monitor* component of Betty is the equivalent of the perception system. It is tailored to understand user actions that include creating, modifying, and deleting concepts and links in the concepts map structure, querying Betty, asking her to explain her answer, and requesting that she take a quiz with the teacher.

The primary component of the agent is its *decision maker* and *memory* subsystems (in a loose way they correspond to the planner and knowledge base, respectively). The decision maker implements the qualitative reasoning mechanisms that can draw inference from concept map structures. The reasoning mechanism is designed to answer queries posed by the student, generate answers to quizzes, and provide explanations for how answers

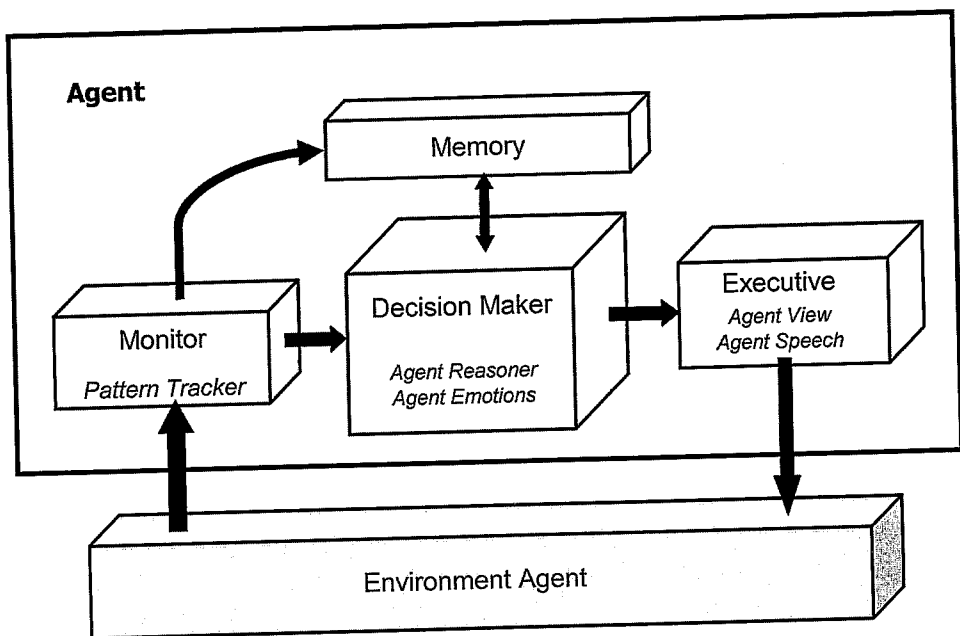


FIGURE 2 Components of agent architecture.

were derived. In addition, the reasoner implements strategies that govern the dialogue process with the user. In some sense, the reasoning mechanisms also play the role of homeostatic control in that they can inform the student about Betty's lack of knowledge in the context of queries that are posed to her.

The *executive* plays the same role as the motor control subsystem for robotics systems. It controls the dialogue mechanisms, and Betty's speech and animation engines. These are primarily used to explain how Betty derives her answer to a question. The first version of the system that we describe next was purely reactive and did not include deliberative planning mechanisms. A second version of the system includes self-regulation strategies, and this requires some planning to determine how Betty should behave and respond to the student's requests.

A second agent, the *mentor*, Mr. Davis, provides hints to Betty and her student teacher on how to improve Betty's performance after she takes a quiz. The mentor agent knowledge base includes a complete concept map of the domain and the qualitative reasoning mechanisms. He also has additional mechanisms to compare the expert map with the student-created map, and to use the differences to provide appropriate feedback. The mentor provides different levels of hints that range from general suggestions to specific feedback on concepts and links the students needs to add to the concept map to get a quiz question correct. Structured templates define the mentor's dialogue structure. We discuss the individual components of Betty's Brain in more detail next.

Teach Betty

Students teach Betty by creating a concept map. A concept map is a collection of concepts and relations between these concepts (Novak 1996). A relation is a unidirectional link connecting two entities. Concepts maps provide an expressive graphic language for creating domain knowledge structures, and this provides students with a means for creating sophisticated structures without getting involved in complex programming tasks (Biswas et al. 2001; Leelawong et al. 2003).

Figure 1 displays an example of a concept map that represents what the student has taught Betty. This map is not a complete representation of all the knowledge in the domain, but merely an example. The labeled boxes correspond to concepts (the labels are concept names), and the labeled links correspond to relations. Students can use three kinds of links, (i) causal, (ii) hierarchical, and (iii) property. Students use property links to embed notes or interesting characteristics of an object in their concept map (e.g., "Fish live by rocks."). Hierarchical links let students establish class structures to organize domain knowledge (e.g., "Fish is a type of animal.").

A causal link specifies an active relationship on how a change in the originating concept affects the destination concept. Two examples of this type of relation are "Plants use carbon dioxide," and "Plants produce dissolved oxygen." The causal relations are further qualified by increase (++) and decrease (--) labels. For example, "use" implies a decrease relation, and "produce" an increase. Therefore, an introduction of more plants into the ecosystem causes a decrease in the amount of carbon dioxide. An increase in the number of plants also causes an increase in dissolved oxygen. When students create a causal link, they are explicitly required to specify whether the link is an increase or a decrease relation.

Query Betty

Students can query Betty about what they have taught her. The system provides a template that students use to create queries for Betty, e.g., *If Concept A increases (decreases) what happens to Concept B?* The query mode uses two primary components: (i) the qualitative reasoning mechanism, and (ii) the explanation mechanism. The reasoning mechanism enables Betty to generate answers to questions from the concept map that the student has taught her. The explanation mechanism enables Betty to produce a detailed explanation of how she generated her answer.

The reasoning mechanism uses a simple chaining procedure to deduce the relationship between a set of connected concepts. To derive the effect of a change (either an *increase or a decrease*) in concept A on concept B, Betty performs the following steps:

Starting from concept A, propagate the effect of its change through all outgoing casual links (i.e., follow the link from concept A to all its adjacent concepts) by pairwise propagation using the relations described in Table 1. This process is repeated for the next set of concepts, which now have an increase/decrease value. Repeated application of this step in a breadth-first manner creates a chain of reasoning through which the change in the source concept (A) propagates to define the change in the destination concept (B).

However, for any concept along a propagation path, if the number of incoming casual links is more than one, the forward propagation stops until all incoming links are resolved. To derive the result from two incoming links, we use the combination algebra defined in Table 2. A "?" in Table 2 implies an unknown change (attributed to the ambiguity of qualitative arithmetic).

If the number of incoming links is three or more, we count the number of changes that fall into the six categories: large ($-_L$), normal ($-$), and small decrease ($-_S$) and small ($+_S$), normal ($+$), and large ($+_L$) increase. The subscripts S and L in Tables 1 and 2 stand for small and large, respectively.

TABLE 1 The Pair-Wise Effects

		Change in Relation					
Change in Entity		$+_L$	$+_L$	$+$	$-$	\neg_L	\neg_L
	$+_L$		$+$	$+_S$	\neg_S	$-$	\neg_L
	$+$		$+_S$	$+_S$	\neg_S	\neg_S	$-$
	$-$		\neg_S	\neg_S	$+_S$	$+_S$	$+$
	\neg_L		$-$	\neg_S	$+_S$	$+$	$+_L$
	\neg_L		\neg_L	$-$	$+$	$+_L$	$+_L$

Combine the corresponding (i.e., small, medium, and large) changes; always subtract the smaller number from the larger. For example, if there is one arc that says *small decrease* (\neg_S), and two incoming arcs that say *small increase* ($+_S$), the result is derived to be a *small increase* ($+_S$). To compute the overall effect, if the resultant value set has all increases or all decreases, we select the largest change. Otherwise, we start at the smallest level of change and combine with the next higher level in succession using the relations defined in Table 2. The overall qualitative reasoning mechanism is a simplified implementation of qualitative process theory (Forbus 1984).

TABLE 2 Integrating Results from Two Paths

		$+_L$	$+_L$	$+_L$	$+$	$+_S$	$?$
	$+_L$		$+_L$	$+$	$+_S$	$?$	\neg_S
	$+$		$+$	$+$	$?$	\neg_S	$-$
	$-$		$+_S$	$?$	$-$	$-$	\neg_L
	\neg_L		$?$	\neg_S	$-$	\neg_L	\neg_L
	\neg_L		\neg_S	$-$	\neg_L	\neg_L	\neg_L

To illustrate the reasoning process, we outline the explanation that Betty generates when she is asked to answer a question: "*If bacteria increase, what happens to animals?*," using the ecosystem concept map shown in Figure 1. As discussed earlier, the qualitative reasoning mechanism employs a breadth-first search to find all paths that lead from the source concept to the destination concept. If there is a single path from the source concept to the destination concept, Betty follows the chain of reasoning step-by-step to illustrate her explanation process. For example, an increase in bacteria will result in more nutrients, and therefore, cause a larger amount of crowded plants. Otherwise, propagation occurs along a path, until it encounters a concept with two or more incoming links. In this case, the effects from all of the links have to be aggregated before forward propagation is resumed. An example is *dissolved oxygen* that has two incoming links.

To structure this process and make it easier for the student to understand Betty's reasoning processes, we break down the explanation into chunks. Betty produces her explanation in a top down fashion. Forward propagation reveals that *bacteria* affect *animals* through *dissolved oxygen* (see Figure 3(a)). Betty summarizes the answer she has derived: "*I'll explain how bacteria affect animals. An increase in bacteria cause dissolved oxygen to decrease a lot, which causes animals to decrease a lot.*" She reports these findings verbally, and illustrates the process in the concept map by animation. The system also includes a *talk log* button. The talk log keeps a record of all previous conversations, and students can access them at any time, to review previous dialogue. Note the process that we have adopted to generate the explanation. Though Betty reasons forward, she explains her answer using a back trace. We have found that students find it easier to follow the reasoning process and the explanation, if it is chunked in this way.

In the next step, Betty builds a more detailed explanation by exploring how bacteria affect dissolved oxygen. Using the reasoning process, Betty has discovered two forward paths from bacteria to dissolved oxygen. One is a direct link, and the second involves a chain of reasoning through the intermediate concepts *nutrients*, *crowded plants*, *sunlight*, and *plants*. Figures 3(c) and (d) illustrate her explanation of how bacteria affect dissolved oxygen using the propagation methods and chain of reasoning through each one of these paths. As a last step, she explains the aggregation step, and reiterates the final answer that an increase in bacteria is likely to cause a large decrease in animals (i.e., macroinvertebrates and fish).

In summary, we designed the query and explanation mechanisms to allow for a dialogue between the student teacher and the teachable agent using a shared representation and reasoning mechanism. The goal was to create an effective teaching environment with feedback that promoted learning. Betty employs animation and speech to explain her thinking to

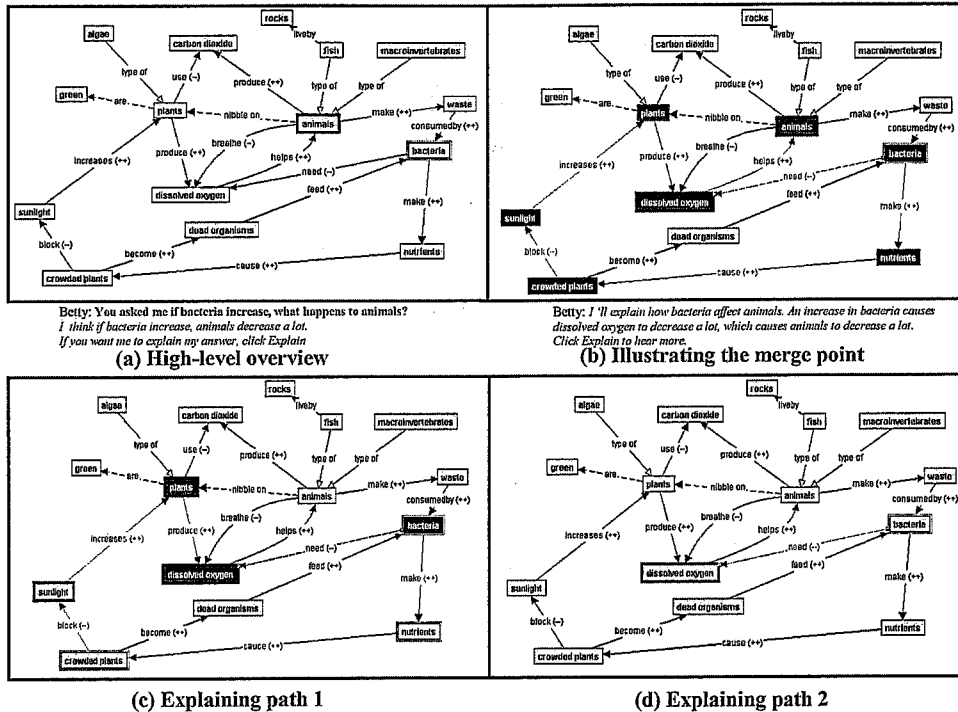


FIGURE-3 Illustrating the structure and animation of the explanation process.

the students. The structure of Betty's explanations is closely tied to the reasoning algorithm. To avoid information overload, the explanation is broken down into segments.

Quiz Betty

The learning environment has an additional mechanism that allows students to assess what they have learned, by having Betty take a quiz, and observing how she performs. The quiz questions, typically written by the system designers and classroom teachers, provide an external assessment mechanism. The quiz interface and a set of quiz questions are illustrated in Figure 4. When Betty takes a quiz, the mentor agent grades the quiz and informs Betty (and the student) if Betty's answers are right or wrong. The mentor also gives hints to help the student debug the concept map. As discussed, the mentor employs a simple mechanism for generating feedback. By overlaying the student's concept map on the expert map he identifies concepts and links that are essential for generating the right answer. He uses this information to generate a hint about a missing concept or link or a misplaced or misdirected link. Typically, Mr. Davis provides a hint for each quiz question that was incorrectly answered.

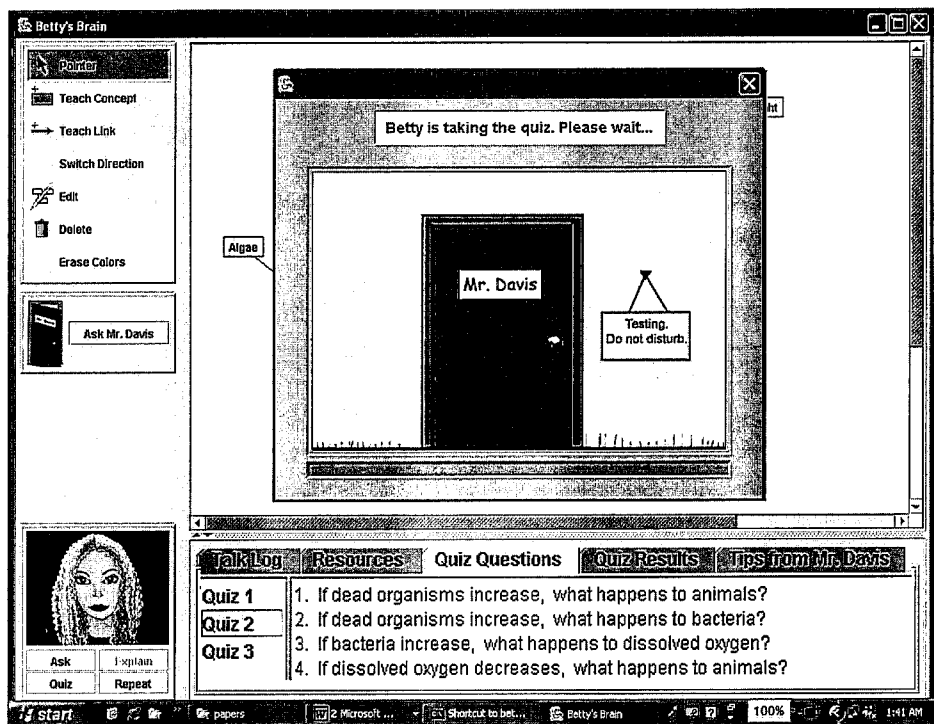


FIGURE 4 Quiz interface and example quiz questions.

In the first study that we describe next, the system implemented three levels of hints. The first time a student got a quiz question wrong, the mentor's agent's hints provided pointers to online resource materials that contained concepts and links relevant to the quiz question. If the answer to the question was still incorrect when Betty took the quiz a second time, the mentor's second hint explicitly mentioned the name of the missing concepts or relations linked to that query. If the student was unable to correct the concept map before Betty took the quiz a third time, the hint provided by the mentor agent was very direct. It told students where to insert missing concepts and links in their concept maps. In some situations, it also told the students about how to correct a misdirected causal link in their current map.

Studies on Betty's Brain (Version 1)

To study the effectiveness of Betty's Brain we conducted an experiment with 50 fifth grade students from a science class in an urban public school located in southeastern United States. We examined the effects of the interactive features of the teachable agent environment, which emulate the feedback that instructors receive from students during teaching. All students

had the opportunity to teach their agent, and we manipulated whether students could query Betty and observe her quiz performance following their teaching efforts. Crossing these variables created four versions of the teachable agent environment: (i) concept map version (no query or quiz), (ii) query version, (iii) quiz version, and (iv) full version (query and quiz).

We hypothesized that having opportunities to query and/or quiz Betty would positively, but differentially, impact students' learning. The query feature helps students debug their own thinking and reasoning in the problem domain. If Betty answers questions in unexpected ways, students know that they need to add to or modify their concept maps. In addition, and perhaps more important, when Betty explains her answers, she makes explicit the process of reasoning along chains of links, and also along multiple paths in a concept map. Therefore, we might expected that students who used the query version of the software would create maps containing more inter-linked concepts. With respect to the quiz condition, we expected that students would become better at identifying important concepts and links to include in their maps, by mapping backward from the quiz questions. We also expected that overall they would produce more accurate concept maps because they had access to feedback on Betty's quiz performance.

The software was used in three sessions of one hour each. At the beginning of session 1, students were introduced to features of the software. They were asked to teach Betty about river ecosystems. In between sessions with Betty, students engaged in independent study to prepare themselves to teach Betty. Reference materials were also available for students to access as needed when preparing to teach and when teaching Betty. Analysis of the quality of the students' maps in terms of the types and accuracy of links suggest several conclusions. It was clear that the students who used the query and quiz mechanisms understood causal relations better than the students who did not. This was reflected in their concept maps, which had a larger proportion of causal links than the teach-only group.

Figure 5(a) shows the ratio of links to concepts in the students' maps, a measure of the interconnectedness of their maps. Overall, query and full students had significantly denser maps than other students. Evidently, having the opportunity to query Betty, which made the reasoning process more explicit, helped students understand the importance of interrelations among concepts in their maps. Figure 5(b) shows the number of valid causal links contained in students' maps. When coding the validity of the links in students' maps, credit was given for links comprising the mentor's expert map as well as for other relevant links related to river ecosystems (determined by our expert raters). Comparisons of the means indicated that by session 3, query students had significantly more valid links in their maps than students in the teach only group. Quiz and full students were

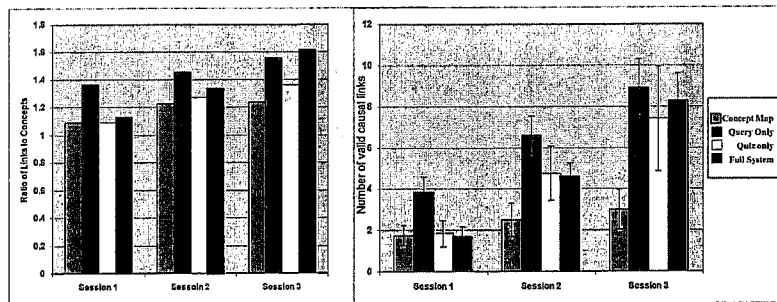


FIGURE 5 (a) Ratio of links to concepts in students' concept maps. (b) Number of valid causal links in students' concept maps.

intermediate and did not differ much from each other. Although the query group had the most valid links, the quiz and full groups had more links from the mentor's map than students in the query group (Leelawong et al., 2002). The data implies that students in the quiz and full groups depended on the quiz and the teacher agent feedback in determining concepts and relations to teach Betty. However, it was not clear how much of a global understanding the quiz-only group had of their overall concept maps. Regardless, this study showed the value of teachable agent paradigm compared to simply representing knowledge. By the third session, students who had an opportunity to query and/or quiz their agents did better overall than students who simply created concepts maps of the domain.

Discussion

Results from the study indicate that both the query and quiz features had beneficial effects on students' learning about ecosystems. Students who had access to the query feature had the most interlinked maps. The query mechanism appears to be effective in helping students develop an understanding of the interrelationships between entities in an ecosystem. Also, the opportunities to quiz their agent helped students to decrease the number of irrelevant concepts, increase the proportion of causal information, and increase the number of expert causal links in their maps. Thus, the quiz feature was effective in helping students decide the important domain concepts and types of relationships to teach Betty. Students inferred, and reasonably so, that if a concept or relationship was in the quiz, it was important for Betty to know.

This notwithstanding, our observations of students during the study suggest that quiz students may have been overly focused on getting the quiz questions correct rather than making sure that Betty (and they themselves) understood the information. We believe that this could partially be attributed

to the nature of the suggestions provided by the mentor agent, which led students to focus on making local changes to their maps instead of paying attention to interdependencies at the level of the (eco)system. Surprisingly, students in the query condition produced as many valid relevant causal links as the conditions with the quiz feature, and without the benefit of quiz feedback. This demonstrates the value of explicitly illustrating the reasoning process (by having Betty explain her answers) so that students understand causal structures.

The full group did not generate significantly higher-quality maps than the quiz and the query groups. An investigation of the activity logs revealed a pattern where students' primary focus was to get the quiz questions correct. After getting Betty to take the quiz, they used the mentor's hints to make corrections to their maps. Very little time (if any) was spent on re-reading the resources to gain more information. The query feature was not used for deep analysis of the concept map; it was primarily used to check whether Betty now answered the particular question correctly after revision. The student then quickly returned to the quiz mode to check on the next question that Betty could not answer correctly. The encouraging observation was that students were motivated and worked to make sure Betty answered all the quiz questions correctly. However, it was not clear that students were making sufficient effort to gain deep understanding of domain knowledge so they could teach Betty better. As noted above, the mentor agent feedback may have inadvertently allowed students to focus on making quick local changes to their maps instead of taking more time to reason globally with their maps.

AN IMPROVED VERSION OF BETTY'S BRAIN

Reflections on the results of the experimental study with version 1 of Betty's Brain led to a rethinking and redesign of the learning environment. A primary concern was the students' focus on getting quiz questions right, without trying to understand the interdependence relations among entities and how they affect the global balance in the river ecosystem structure. As discussed earlier, we realized that feedback from the system (both from the mentor and Betty) had to be improved to facilitate better learning. Further, in exit interviews, students emphasized that they would have liked Betty to be more active and exhibit characteristics of a good student during the teaching phase (Davis et al. 2001). Several students suggested that Betty should be more interactive, e.g., "*react to what she was being taught, and take the initiative and ask more questions on her own,*" and "*do some sort of game or something and make it more interactive.*" Consistent with this idea, we note that the current version of Betty is passive and only responds when asked questions. We believe that to create a true learning-by-teaching environment,

Betty needs to better demonstrate qualities of human students. A tutor gains deeper understanding from interactions with a tutee (Chi et al. 2000; Cole et al. 1999). These interactions can include answering questions, explaining materials, and discovering misconceptions. Betty should be designed to benefit her users in the same fashion.

Version 1 of *Betty's Brain* provided online resources that students could view to learn and clarify their understanding of domain material. There was some feedback from the mentor agent, but it was mainly in the form of suggestions to correct the concept map after Betty had taken a quiz. We would have liked to have seen more use of the online resources when the student was teaching Betty, or reflecting on her quiz performance, especially because our project deals with a unique situation. The students as teachers are novices in the domain. They are also novices at teaching.

An improved learning by teaching environment should incorporate mechanisms to assist students in all phases of the teaching process: *preparation*, *teaching*, and *monitoring* (McAlpine 1999). The challenge was to carefully redesign the learning environment to provide appropriate scaffolds and proper feedback mechanisms to help students overcome their initial difficulties in learning about the domain and their figuring out how to teach Betty well. This led to the reimplementation of a number of components in the learning environment. For one, we restructured the online resources to emphasize the concepts of interdependence and balance and the three primary cycles that govern ecosystem behavior: (i) the oxygen cycle, (ii) the food chain, and (iii) the waste cycle. An advanced keyword search technique allows students to access paragraphs of text with the occurrences of a selected keyword or a pair of selected keywords highlighted.

Changes were also made to the mentor agent. In the previous version, the mentor made suggestions on how to correct specific errors in the concept map after Betty took a quiz. As discussed, this led to students making local changes to their concept maps without trying to gain a proper understanding of the relations between domain concepts. We decided that in the new version of the system, the mentor, Mr. Davis, would direct the student to study more about interdependence among concepts, and how this interdependence leads to "chains of reasoning," i.e., an increase or decrease in a concept can affect a number of other concepts through a sequence of propagations. He directed students to study and reflect on relevant sections in the resources, rather than suggesting what changes to make to their concept maps. Like before, the mentor provides levels of feedback. His initial comments are general, but they become more specific (e.g., "You may want to study the role of bacteria in the waste cycle") if errors persist, or the student seeks help. In addition, Mr. Davis provided more metacognitive feedback by providing information on how to be a *better learner* (e.g., he would point out the importance of goal setting, understanding chains of dependencies, and self-assessing one's

knowledge while learning) and how to be a *better teacher* (e.g., by making high-level suggestions about the representational and reasoning structures, and asking the student to assess Betty's understanding of what they taught her by getting her to answer relevant queries and studying her responses).

Betty's agent persona was also redesigned to incorporate self-regulation strategies proposed by Zimmerman (1989). These include metacognitive strategies like monitoring, assessing, goal setting, seeking assistance, and reflecting on feedback, all of which we believe can aid the learning and teaching tasks. We exploited some of these strategies to drive Betty's interactions with her student teacher. For example, when the student is building the concept map for Betty, she occasionally demonstrates how to derive the effects of the change in an entity on other entities through a chain of reasoning. She may query the user, and sometimes remark (right or wrong) that the answer she is deriving does not seem to make sense. The idea of these spontaneous prompts is to get the student to reflect on what they are teaching, and perhaps, like a good teacher check on their tutee's learning progress. At other times, Betty may directly suggest to the students that they need to query her to ensure that she can reason correctly with the current concept map. At times, Betty 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.

In the present version of the system, Betty directly discusses the results with the student teacher by reporting on (i) her thoughts of her performance on the particular quiz: She may say that she is happy to see her performance has improved, or express disappointment that she failed to answer a question more than once, and (ii) the mentor's comments on Betty's performance in the quiz, such as: *"Hi, I'm back. I'm feeling bad because I could not answer some questions on the quiz. Mr. Davis (the mentor) said that I should have studied about the various entities that participate in the waste cycle."*

We believe self-regulation strategies provide the right scaffolds to help students learn about a complex domain, while also promoting deep understanding, transfer, and lifelong learning. All this is achieved in a constructivist exploratory environment, with the student primarily retaining the locus of control. Only when students seem to be hopelessly stuck, does Mr. Davis intervene with specific help. Next, we present a multiagent architecture that provides a more efficient implementation of the learning-by-teaching system with self-regulation strategies.

A New Multiagent Architecture for Betty's Brain

From a learning system viewpoint, a multiagent architecture was developed to overcome drawbacks of the previous version of Betty's Brain, and to introduce the new features to promote inquiry-based, self-regulated learning that we have discussed earlier. The software system was redesigned

to modularize the various functions required in the system, and to systematically introduce the notion of interactions among the agents (Ferber 1999).

The current multiagent system, illustrated in Figure 6, uses four agents: the teachable agent, the mentor agent, and two auxiliary agents: the student agent and the environment agent. The explicit presence of the last two agents in the Betty's Brain environment is primarily to establish a standardized communication protocol among all agents that participate in the system. In the future, this will provide greater flexibility to move agents from one scenario to another. The student agent is not a person. It provides the interface for the student-teacher to communicate with the teachable agent, Betty, and the mentor agent, Mr. Davis. The environment agent, which acts as a "facilitator" (Finin and Fritzson 1994) is in essence a medium through which all of the agents communicate with each other and get to observe the global state of the system. This 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, and (iii) translates information between vocabularies to match an agent interface.

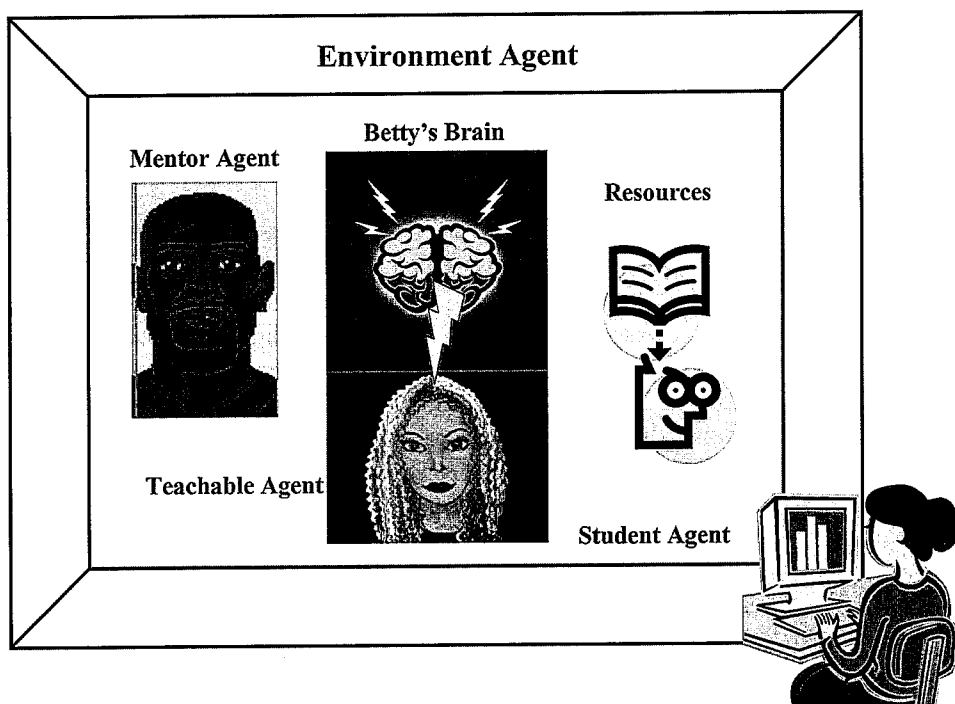


FIGURE 6 Multiagent architecture for Betty's Brain.

The system uses a variation of the FIPA ACL agent communication language (Labrou et al. 1999). 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 the other agents. 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 which decides the modality with which to communicate this decision to the environment.

Experimental Studies with the Multiagent System

The experimental study conducted in the second year compared three different versions of the system. All three groups had access to identical resources on river ecosystems, the same quiz questions, and the same access to the mentor agent, Mr. Davis. In the first system, students did not teach Betty. Instead, they were taught by the mentor agent, Mr. Davis, who asked them to construct concept maps to answer three sets of quiz questions that were designed to meet curricular guidelines. When students submitted their maps for a quiz, Mr. Davis, playing the role of a tutor, provided directed feedback to help the students correct errors in the quiz answers. We called this the Intelligent Tutoring System (ITS) version of the system. The two other groups were told to teach Betty and help her pass a test so she could become a member of the school science club. Both groups had access to the query and quiz features. In one of the two versions of the learning by teaching systems, which we call the baseline learning by teaching (LBT) system, students could query Betty as they were teaching her, ask Betty to take a quiz after they taught her, and, if there were errors in the quiz answers, Mr. Davis provided the same feedback as the ITS system. The only difference was that feedback was directed to Betty because she took the quiz. The second learning by teaching system had the new, more responsive Betty with self-regulated behavior. In addition, the mentor agent was designed to provide a wide variety of help that included information on how to be better learners and how to be better teachers. Like before, Mr. Davis could also provide feedback on domain knowledge concepts. But this group had to explicitly query Mr. Davis to get any feedback. We called this version of the system the self-regulated learning (SRL) system. The SRL condition was set up to develop more active learners by

promoting the use of self-regulation strategies. The ITS condition was created to contrast learning by teaching environments from tutoring environments.

There were two primary research questions we set out to answer were:

1. Are learning-by-teaching environments more effective in helping students to learn independently and gain deeper understanding of domain knowledge than pedagogical agents?
2. Does the self-regulated learning component enhance learning in learning-by-teaching environments?

Self-regulated learning should be an effective framework for providing feedback because it promotes the development of higher-order cognitive skills (Pintrich and DeGroot 1990) and it is critical to the development of problem-solving ability (Novak 1996). In addition, cognitive feedback is more effective than outcome feedback for decision-making tasks (Moreno and Mayer 2002). Cognitive feedback helps users monitor their learning needs (achievement relative to goals) and guides them in achieving their learning objectives (cognitive engagement by applying tactics and strategies).

Experimental Procedure

Students from two fifth grade classrooms were divided into three equal groups of 15 students each using a stratified sampling method. Stratification was based on students' standard achievement test scores in mathematics and language. The students worked on a pretest with twelve questions before they were separately introduced to their particular versions of the system. The three groups worked for six 45-minute sessions over a period of three weeks to create their concept maps. All groups had access to the online resources while they worked on the system.

At the end of the six sessions, every student took a posttest that was identical to the pretest. Two other delayed posttests 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 transfer test*, where they were asked to construct a concept map and answer questions about the land-based nitrogen cycle. Students had not been taught about the nitrogen cycle, so they would have to learn from resources during the transfer phase.

Results

In this study, we focus on the results of the two delayed tests, and the conclusions we can draw from these tests on the students' learning

processes. As a quick review of the initial learning, students in all conditions showed improved performance from pre- to posttest on their knowledge of interdependence (p 's < .01, paired T-tests). However, there was no improvement in their understanding of ecosystem balance. There were few differences between the conditions in terms of the quality of their final maps (the LBT and SRL groups showed better grasp of the role of bacteria in processing waste in their post test answers). However, there were notable differences in their use of the system during the initial learning phase.

Figure 7 shows the average number of resource, query, and quiz requests per session by the three groups. It is clear from the plots that the SRL group made a slow start as compared to the other two groups. This can primarily be attributed to the nature of the feedback, i.e., the ITS and LBT groups received specific content feedback after a quiz, whereas the SRL group tended to receive more generic feedback that focused on self-regulation strategies. Moreover, in the SRL condition, Betty would refuse to take a quiz unless she felt the user had taught her enough, and prepared her for the quiz by asking questions. After a couple of sessions, the SRL group showed a surge in map creation and map analysis activities, and their final concept maps were comparable to the other groups.

It seems the SRL group spent their first few sessions in learning self-regulation strategies, but once they learned them their performance improved significantly. Table 3 presents the mean number of expert concepts and expert causal links in the student maps for the delayed memory test. Results of an ANOVA test on the data, with Tukey's LSD to make pairwise comparisons, showed that the SRL group recalled significantly more links that were also in the expert map (which nobody actually saw).

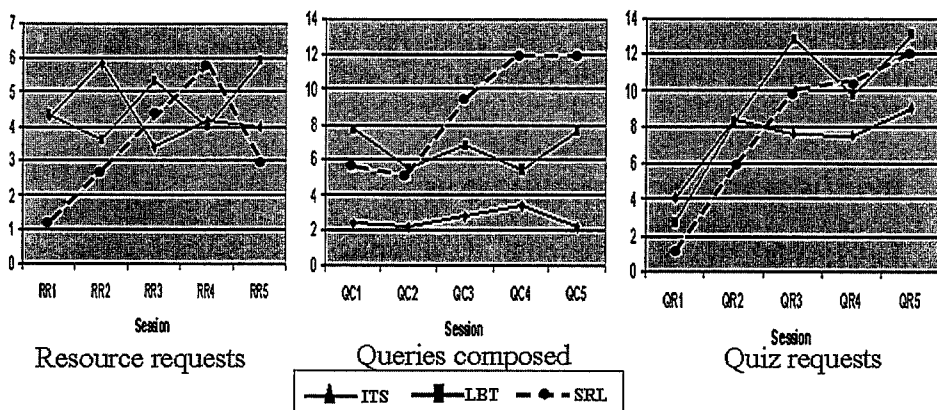


FIGURE 7 Session by session data for study 2.

TABLE 3 Results of the Memory Test

Student map included:	SRL mean (sd)	LBT mean (sd)	ITS mean (sd)
Expert concepts	6.7 (.6)	6.4 (.5)	5.8 (.6)
Expert causal links	3.3 ^a (.6)	1.7 (.6)	2.0 (.6)

^aSignificantly greater than LBT, $p < .05$;

TABLE 4 Results of the Transfer Study

Student map included:	SRL mean (sd)	LBT mean (sd)	ITS mean (sd)
Expert concepts	6.1 ^a (.6)	5.2 (.5)	4.1 (.6)
Expert causal links	1.1 ^{ab} (.3)	0.1 (.3)	0.2 (.3)

^aSignificantly greater than ITS, $p < .05$;

^bsignificantly greater than LBT, $p < .05$

We thought that the effect of SRL would not be to improve memory, but rather to provide students with more skills for learning subsequently. When one looks at the results of the transfer task in the test on preparation for future learning, the differences between the SRL group and the other two groups are significant. Table 4 summarizes the results of the transfer test, where students read resources and created a concept map for the land-based nitrogen cycle which they had not studied previously. The mentor agent's only provided feedback on the correctness of the answers to the quiz questions, but no hints on how the students could improve Betty's performance. All three groups received the same treatment. There are significant differences in the number of expert concepts in the SRL versus ITS group maps, and the SRL group had significantly more expert causal links than the LBT and ITS groups. The effects of teaching self-regulation strategies had an impact on the students' abilities to learn a new domain, two months later.

DISCUSSION AND CONCLUSIONS

The second set of experimental result brings out a number of interesting issues. First, since the tutored students were focused on the task of creating concept maps to answer specific questions, they initially outperformed the two learning-by-teaching groups. The SRL group seemed to be the slowest of the three, but, as discussed earlier, this can be attributed to the fact that this group spent the first couple of sessions in learning self-regulation strategies. Once they understood these strategies,

their performance improved considerably, and at the end of the initial learning period, all three groups showed about equal performance. This was measured in terms of the quality of their concept maps.

The three groups had about equal performance in their memory tests, but the SRL group demonstrated better abilities to learn and understand new material by outperforming the ITS and LBT groups in the far transfer test. This result is important in that it demonstrates the significance of SRL strategies in aiding understanding and transfer in learning by teaching environments. Students in all three groups demonstrated the same learning performance in traditional learning tasks, but the SRL group demonstrated better ability to learn new material without the scaffolds that were provided in the first part of the study. We believe that the difference between the SRL and the other two groups would have been more pronounced if the transfer test study had been conducted over a longer period of time.

In summary, we have developed a new form of pedagogical agent and a learning environment that goes much beyond the notion of virtual tutors and traditional learning by teaching systems. In future work, we will extend the concept map representation and reasoning mechanisms to accommodate reasoning over time, feedback effects, and cycles of behavior, all of which are common phenomena in natural processes. It is possible that other extensions to this environment, such as getting students to create Betty agents that compete with each other in game shows, or projecting a set of student concept maps in front of the class so students may compare and contrast maps will add to learning and motivation (Schwartz, Bransford, and Sears, in press). Finally, an exciting direction would be to combine teachable agents with game environments. The goal will be to create a sequence of challenging knowledge tasks. Students will need to learn and teach different agents, each with their own representational and reasoning structures to solve these challenges. The environment can be structured to allow for single players to learn complex problem solving through multiple forms of interaction further social interactions through collaborative problem solving, and create competitive environments where students can compete with each other through their teachable agents.

REFERENCES

- Arkin, R. C. 1991. Reactive control as a substrate for telerobotic system. *IEEE Aerospace and Electronics Systems Magazine* 6(6):24-31.
- Artzt, A. F. and E. Armour-Thomas. 1999. Cognitive model for examining teachers' instructional practice in mathematics: A guild for facilitating teacher reflection. *Educational Studies in Mathematics* 40(30):211-335.
- Ball, G., D. Ling, D. Kurlander, J. Miller, D. Pugh, T. Skelly, A. Stankosky, D. Theil, M. Van Dantzich, and T. Wax. 1997. Lifelike computer characters: The persona project at Microsoft research. In: *Software Agents*, ed. J. M. Bradshaw, 191-222. Menlo Park, CA: AAAI/MIT Press.

- Bargh, J. A. and Y. Schul. 1980. On the cognitive benefits of teaching. *Journal of Educational Psychology* 72(5):593-604.
- Biswas, G., D. L. Schwartz, and J. D. Bransford. 2001. Technology support for complex problem solving: Form SAD environments to AI. In: *Smart Machines in Education*, eds. K. Forbus and Feltovich, 71-98, Menlo Park, CA: AAAI Press.
- Biswas, G., K. Leelawong, K. Belyne, K. Viswanath, N. J. Vye, D. L. Schwartz, and J. Davis. 2001. Incorporating self regulated learning techniques into learning by teaching environments. In *Proceedings of the 20th Annual Meeting of the Cognitive Science Society*, pages 120-125, Chicago, Illinois, USA.
- Bransford, J. D., J. J. Franks, N. J. Vye, and R. D. Sherwood. 1989. New approaches to instruction: Because wisdom can't be told. In: *Similarity and Analogical Reasoning*, eds. S. Vosniadou and A. Ortony, 470-497. New York: Cambridge University Press.
- Bransford, J. D. and D. L. Schwartz. 1999. Rethinking transfer: A simple proposal with multiple implications. In: *Review of Research in Education*, eds. A. Iran-Nejad and P. D. Pearson, vol 24, 61-101. Washington, D.C.: American Educational Research Association.
- Bransford, J. D., A. L. Brown, and R. R. Cocking, eds. 2000. *How People Learn*. Washington, DC: National Academy Press.
- Burton, R. B. and J. S. Brown. 1982. An investigation of computer coaching for informal learning activities. *Intelligent Tutoring Systems*. London: Academic Press.
- Cassell, J. 2004. Towards a model of technology and literacy development: Story listening system. *Journal of Applied Developmental Psychology* 25(1):75-105.
- Chan, T.-W. and C.-Y. Chou. 1997. Exploring the design of computer supports for reciprocal tutoring. *International Journal of Artificial Intelligence in Education* 8:1-29.
- Chi, M. T. H., N. De Leeuw, C. Mei-Hung, and C. Levancher. 1994. Eliciting self explanations. *Cognitive Science* 18:439-477.
- Chi, M. T. H., S. A. Siler, H. Jeong, T. Yamauchi, and R. G. Hausmann. 2001. Learning from human tutoring. *Cognitive Science* 25(4):471-533.
- Cohen, P. A., J. A. Kulik, and C.-L. Kulik. 1982. Educational outcomes of peer tutoring: A meta-analysis of findings. *American Educational Research Journal* 19(2):237-248.
- Cole, R., D. W. Massaro, B. Rundle, K. Shobaki, J. Wouters, M. Cohen, J. Beskow, D. Stone, P. Connors, A. Tarachow, and D. Solcher. 1999. New tools for interactive speech and language training: Using animated conversational agents in the classrooms of profoundly deaf children. Paper presented at the *ESCA/SOCRATES Workshop on Method and Tool Innovations for Speech Science Education*, London, UK.
- Crews, T. R., G. Biswas, S. Goldman, and J. Bransford. 1997. Adventureplayer: A microworld anchored in a macrocontext. *International Journal of AI in Education* 8:142-178.
- Davis, J. M., K. Leelawong, B. Bodenheimer, G. Biswas, N. Vye, and J. Bransford. 2003. Intelligent user interface design for teachable agent systems. In *Proceedings of the International Conference on Intelligent User Interfaces*, pp. 26-34. Miami, Florida, USA. The Association for Computing Machinery.
- Dayer, P. 1996. *Le Tutorat Virtuel: Appendre en Construisant un Didacticiel*. Unpublished Graduate Dissertation. University of Geneva, Geneva.
- Ferber, J. 1999. *Multi-agent Systems: An Introduction to Distributed Artificial Intelligence* 1st ed. Harlow, UK: Addison-Wesley Professional.
- Finin, T., Y. Labron, and J. Mayfield. 1997. KQML as an agent communication language, in *Software Agents*, ed. J.M. Bradshaw, pp. 291-316. Menlo Park, CA: AAAI/MIT Press.
- Forbus, K. 1984. Qualitative process theory. *Artificial Intelligence* 24:85-168.
- Graesser, A. C., N. Person, and J. Maglian. 1995. Collaborative dialog patterns in naturalistic one-on-one tutoring. *Applied Cognitive Psychologist* 9:359-387.
- Hietala, P. and T. Niemirepo. The competence of learning companion agents. *International Journal of Artificial Intelligence in Education* 9:178-192.
- Huffman, S. B. and J. E. Laird. 1995. Flexibly instructable agents. *Journal of Artificial Intelligence Research* 3:271-324.
- Kafai, Y. and I. Harel. 1999. Learning through design and teaching: Exploring social and collaborative aspects of constructionism. In: *Constructionism*, eds. I. Harel and S. Papert, pp. 111-140. Norwood, NJ: Ablex.

- Kinchin, I. M. and D. B. Hay. 2000. How a qualitative approach to concept map analysis can be used to aid learning by illustrating patterns of conceptual development. *Educational Research* 42(1):43–57.
- Labrou, Y., T. Finin, and Y. Peng. 1999. Agent communication languages: The current landscape. *Intelligent Systems* 14(2).
- Lajoie, S. P. 1993. Computer environments as cognitive tools for enhancing learning. In: *Computer as Cognitive Tools*, eds. S. P. Lajoie and S. J. Derry, Hillsdale NJ: Lawrence Erlbaum Associates, pp. 261–288.
- Leelawong, K., J. Davis, N. Vye, G. Biswas, D. Schwartz, K. Belyne, T. Katzlberger, and J. Bransford. 2002. The effects of feedback in supporting learning by teaching in a teachable agent environment. In *Proceedings of the Fifth International Conference of the Learning Sciences*, 245–252, Seattle, Washington, USA.
- Leelawong, K., et al. 2003. Teachable agents: Learning by teaching environments for science domains. In *Proceedings of the 15th Innovative Applications of Artificial Intelligence Conference*, pages 109–116. Acapulco, Mexico.
- Lieberman, H. and D. Maulsby. 1996. Instructible agents: Software that just keeps getting better. *IBM Systems Journal* 35(3/4).
- Michie, D., A. Paterson, and J. E. Hayes. 1989. Learning by teaching. In *Proceedings of the Second Scandinavian Conference on Artificial Intelligence (SCAI)*, pages 413–436. Tampere, Finland. IOS Press.
- Moreno, R. and R. E. Mayer. 2002. Learning science in virtual reality multimedia environments: Role of methods and media. *Journal of Educational Psychology* 94:598–610.
- Novak, J. D. 1996. Concept mapping as a tool for improving science teaching and learning. In: *Improving Teaching and Learning in Science and Mathematics*, eds. D. F. Treagust, R. Duit, and B. J. Fraser, 32–43. London: Teachers College Press.
- Nichols, D. M. 1994. Intelligent Student Systems: An Application of Viewpoints to Intelligent Learning Environments. Unpublished Ph.D. Dissertation, Lancaster University, Lancaster, UK.
- Obayashi, F., H. Shimoda, and H. Yoshikawa. 2000. Construction and evaluation of CAI system based on 'Learning by Teaching to Virtual Student.' In *Proceedings of the World Multiconference on Systemics, Cybernetics and Informatics*. Orlando, Florida, vol. 3, pp. 94–99.
- Palincsar, A. S. and A. L. Brown. 1984. Reciprocal teaching of comprehension-fostering and comprehension monitoring activities. *Cognition and Instruction* 1:117–175.
- Srinivas, P., J. E. Greer, and G. I. McCalla. 1991. Learning by teaching. In *The International Conference on Learning Sciences* Illinois, USA, pp. 357–363.
- Pintrich, P. R. and E. V. DeGroot. 1990. Motivational and self-regulated learning components of classroom academic performance. *Journal of Educational Psychology* 82:33–40.
- Ponder, G. K. and J. Kelly. 1997. Evolution, chaos, or perpetual motion? A prospective trend analysis of secondary science curriculum advocacy, 1955–1994. *Journal of Curriculum and Supervision* 12(3):238–245.
- Quinlan, J. R. 1986. Induction of decision trees. *Machine Learning* 1(1):81–106.
- Reeves, B. and C. Nass. 1996. *The Media Equation: How People Treat Computers, Televisions and New Media Like Real People and Places*. Cambridge: Cambridge University Press.
- Repenning, A. and T. Sumner. 1995. Agentsheets: A medium for creating domain oriented visual languages. *Computer* 28:17–25.
- Rickel, J. and W. L. Johnson. 1998. STEVE: A pedagogical agent for virtual reality (video). In *Proceedings of the Second International Conference on Autonomous Agents*, Minneapolis/St. Paul, MN, USA, May 1998. ACM Press.
- Schwartz, D. L. and T. Martin. 2004. Inventing to prepare for learning: The hidden efficiency of original student production in statistics instruction. *Cognition & Instruction* 22:129–184.
- Schwartz, D. L., K. P. Blair, G. Biswas, K. Leelawong, and J. Davis. Animations of thought: Interactivity in the Teachable Agents Paradigm. To appear in R. Lowe and W. Schnotz (eds.) *Learning with Animation: Research and Implications for Design*. UK: Cambridge University Press.
- Schwartz, D. L., J. D. Bransford, and D. Sears. Efficiency and Innovation in Transfer. To appear in J. Mestre (ed.) *Transfer of Learning from a Modern Multidisciplinary Perspective*. CT: Information Age Publishing.
- Shimoda, T. A., B. Y. White, and J. R. Frederiksen. 2002. Student goal orientation in learning inquiry skills with modifiable software advisors. *Science Education* 86(2):244–263.

- Smith, D. C., A. Cypher, and J. Spohrer. 1997. Programming agents without a programming language. In *Software Agents* ed. J. M. Bradshaw, Menlo Park, 165–190. CA: AAAI/MIT Press.
- Stoyanov, S. and P. Kommers. 1999. Agent-support for problem solving through concept-mapping. *Journal of Interactive Learning Research* 10(3/4):401–42.
- Wenger, E. 1987. *Artificial Intelligence and Tutoring Systems*. Los Altos, CA: Morgan Kaufmann Publishers.