Bodenheimer, B., Williams, B., Kramer, M. R., Viswanath, K., Balachandran, R., Belynne, K., & Biswas, G. (2009). Construction and Evaluation of Animated Teachable Agents. *Educational Technology & Society, 12* (3), 191–205.

Construction and Evaluation of Animated Teachable Agents

Bobby Bodenheimer, Betsy Williams, Mattie Ruth Kramer, Karun Viswanath, Ramya Balachandran, Kadira Belynne and Gautam Biswas
Department of Computer Science and Electrical Engineering, Vanderbilt University, Nashville, USA // Tel: +1 615 322-3555 // Fax: +1 615 343-5459 // bobby.bodenheimer@vanderbilt.edu

ABSTRACT

This article describes the design decisions, technical approach, and evaluation of the animation and interface components for an agent-based system that allows learners to learn by teaching. Students learn by teaching an animated agent using a visual representation. The agent can answer questions about what she has been taught and take quizzes. The agent's performance provides students feedback and motivation to learn, and the overall interaction helps students learn for themselves. The teachable agent uses speech and animation to communicate with the student, and sometimes expresses emotion through her spoken dialog and facial expressions. The technical approach is novel in that it provides a system for creating and customizing an animated agent and associated content in a fast and efficient manner that does not require specialized modeling or animation skills. We evaluate both our interface design and the effects of animated agent on students in a U.S. public school aged 9-11. Results show that the both the new interface and the presence of an animated agent promote positive learning experiences.

Keywords

Teachable agent, Learning environment, Pedagogical agent, Animated agent

Introduction

This paper describes the educational technology and interface behind a model-free, customizable animated agent and its evaluation in a classroom setting for learners aged 9-11 (U.S. fifth grade). The agent, called a teachable agent, is a component of a learning system that implements the principles of learning by teaching. It is well-documented in the educational literature that the process of teaching others can be a powerful method of learning, e.g., (Armis 1983). Studies have shown 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 (Bargh and Schul 1980). Similarly, literature on tutoring suggests that tutors benefit as much from tutoring as their tutees (Graesser et al. 1995; Chi et al. 2001). Biswas et al. (2001) report that students preparing to teach stated that the responsibility of teaching forced them to a deeper understanding of the material; other students focused on the importance of having a clear conceptual organization of the materials.

A teachable agent (TA) environment (Biswas et al. 2005; Nichols 1994; Ramirez-Uresti and du Boulay 2004) is one where learners explicitly teach and directly receive feedback about their teaching through interactions with a computer agent. In our TA system, called Betty's Brain, students learn by teaching a computer agent called Betty to solve problems and answer questions in a variety of scientific domains using graphical cause and effect structures. Unlike previous TA work (e.g., (Michie et al. 1989; Nichols 1994; Ramirez-Uresti and du Boulay 2004)), Betty has no a priori knowledge, only an ability to reason (Davis et al. 2003; Leelawong and Biswas 2008; Viswanath et al. 2004). The agent is explicitly taught knowledge by learners, who are learning and organizing their own knowledge as they teach. Teachable agents are thus similar to pedagogical agents (Baylor and Ryu 2003; Baylor 2005; Graesser et al. 1995; Towns et al. 1998) in the sense that the agent helps create a positive learning experience and facilitates meaningful interaction between the computer and the learner (as shown by, for example, (Craig et al. 2002; Moreno and Mayer 2000; Lester et al. 1999)). However, the pedagogical agents cited above are primarily demonstrative and active tutors of knowledge, whereas a teachable agent positions itself as a learner. The effectiveness of the teachable agent environment in producing learning-related outcomes has been reported in (Davis et al. 2003; Biswas et al. 2005; Tan et al. 2006; Leelawong and Biswas 2008).

The particular goals of this article are twofold. First, we present an analysis of the user interface of the teachable agent system as it relates to its effectiveness in promoting learning related outcomes. Part of the user interface is an animated videorealistic character that serves as the teachable agent. We present the novel technical machinery that allows the construction of such an agent. We use the term videorealistic to mean that the agent is not cartoon-like (a cartoon image was used in earlier studies, e.g., (Biswas et al. 2005; Davis et al. 2003)), but looks closer to a photograph of a person. The structure of our machinery does not use an underlying graphical model, as is commonly used, e.g., (Cole et al. 1999; Elliott et al. 1999), but is based only on image data. This approach can be used to author content easily and customize the teachable agent system as desired. Authoring animation and customizing the appearance and emotional expressions of the agent is typically an expensive process, one which our method seeks to make easier. The goal of this approach is thus to create an animated agent that can reliably convey expressions and emotions that we believe are conducive to learning, in a particular environment, and for a particular class of learners. Animated teachable agents have the potential to affect learning by providing interactions that enhance the constructivist learning experience and increasing motivation through recognized social interactions (Baylor et al. 2004).

More particularly, we explicate these goals and answer two specific questions: (a) does the presence of an animated agent in a learning system create a more positive learning experience than the learning system without the animation; and (b) how does the representation of the agent (its appearance, style of emotion, and animated content) affect the learning experience. While prior studies have asked these questions for college-age learners (e.g., (Ryu and Baylor 2005)), less attention has been devoted to these issues for elementary and middle school children. This factor is critical, since affordances of a system that benefit advanced learners may not create a positive experience for 9-11 year olds (Viswanath et al. 2004). Quick customizability and a model-free representation in the animation engine of the underlying agent can allow us to assess these conditions better. This paper addresses only the first question above, and does so in the context of the Betty's Brain system.

This paper discusses studies where students teach their agent about the domain of pond and river ecosystems. In Tennessee, students in public schools age 9-11 learn about ecosystems as part of their science curriculum, including the major levels of biological classification, the food chain, photosynthesis, and the waste cycle. The aim of the Betty's Brain system is to aid students in learning about entities and interdependence in such ecosystems by studying the causal relationship between pairs of entities, e.g., fish breathe dissolved oxygen, and how these changes propagate to other entities to establish chains of events. We describe these items in greater detail below.

Background

An automated learning by teaching system requires a representation scheme for learners to create their knowledge structure as a part of the teaching process. Since the primary users are students aged 9-12 (5th and 6th grade students in U.S. "middleschools") who are solving complex problems, this representation has to be intuitive but sufficiently expressive to help these students create, organize, and analyze their problem solving ideas. We adapted and extended a widely accepted technique for constructing knowledge, a concept map (Novak 1996; Spiro and Jehng 1990), as our visual representation scheme for the teaching task.

Concept maps provide a mechanism for structuring and organizing knowledge into hierarchies, and allow the analysis of phenomena such as cause-effect relations (Leelawong et al. 2001; Kinchin et al. 2000; Stoyanov and Kommers 1999; Novak 1998; Novak 1996). Moreover, our studies have shown that students relate to the intelligent software agents, particularly because the agents use the concept map to generate answers to questions, and explain how they derived their answers (Biswas et al. 2005; Leelawong and Biswas 2008). In all of our studies, students have not had problems using the concept map interface. Therefore, concept maps provide an excellent representation that helps students teach their agent and monitor their learning through the teaching process.

The Concept Map

A concept map is a collection of concepts and the relationships between these concepts that collectively represents domain knowledge (Novak 1996). A partial concept map in the domain of river ecosystems is shown in Figure 1. The labeled boxes correspond to entities (the labels are entity names), and the labeled links correspond to relations between the entities. The arrow indicates the direction of the relation, and its name appears by the arrow. The parenthesized phrase indicates the relation type.

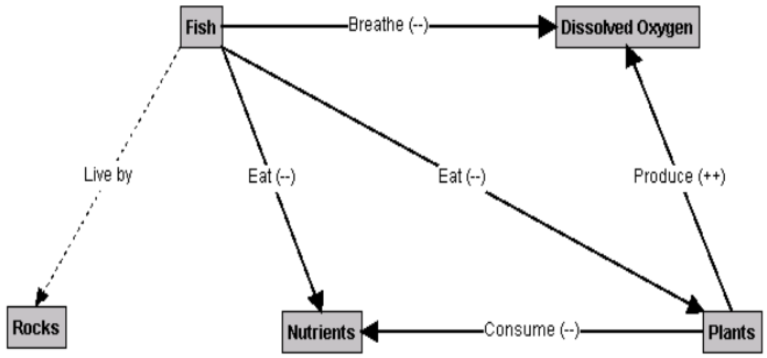

Figure 1: Example concept map for a river ecosystem

In our environment, concepts are entities that are of interest in the domain of study. Common entities in a river ecosystem are fish, plants, bacteria, dissolved oxygen, carbon dioxide, algae, and waste. Relations are unidirectional, binary links connecting two entities. They help to categorize groups of objects or express causal relations between them.

In the current implementation of this domain knowledge, learners can use three kinds of relations to build a concept map: (i) cause-effect, (ii) descriptive, and (iii) type-of. The primary relation learners use to describe relations between entities is the cause-effect relation, such as "Fish eat Plants" and "Plants produce Dissolved oxygen". The causal relations are further qualified by increase ('++') and decrease ('–') labels. For example, "eat" implies a decrease relation, and "produce" implies an increase. Therefore, an introduction of more fish into the ecosystem causes a decrease in plants, while an increase in plants causes an increase in dissolved oxygen.

The descriptive link represents a broader relation between two entities, but the change in one entity does not cause a change in the other entity. An example is the relation "Fish live by Rocks". The "live by" relation is descriptive, because an increase or decrease in fish does not directly change the amount of rocks. More complex forms of the "descriptive" relation, e.g., "Plants need Sunlight to produce Dissolved Oxygen" are conditional causal relations and have been implemented in some versions of the system, but not in the one discussed in this article.

Type-of relations let learners establish class structures to organize the domain knowledge. Consider an example where learners deal with a variety of fish, such as trout, bass, blue gill, and catfish. All of these fish types breathe dissolved oxygen and eat plants. To simplify the knowledge construction process, learners can first create the entity "Fish", and express the "Fish eat Plants" and "Fish breathe Dissolved Oxygen" relations. Then, they can create individual fish entities, such as "trout" and "bass", and link them to the "Fish" entity using "type-of" links. All relations associated with the entity "Fish" are inherited by these individual types unless they are over-ridden by more specific links (Russell and Norvig 1995).

Description of Betty's Brain

This section presents the user interface as described by (Davis et al. 2003), which did not contain an animated agent. The next section presents an analysis evaluating the user interface of the teachable agent system. It analyzes the various trade-offs involved in determining which features should be present in the user interface and how the various components should be designed, particularly with the idea of adding animation. Some of these refinements were suggested by (Viswanath et al. 2004) in a preliminary study on the efficacy of the user interface. All refinements and changes will be evaluated through studies on public school students age 9-11 (U.S. fifth grade), a target audience for this system.

Figure 2 illustrates the Betty's Brain interface. The system possesses multimedia capabilities. Betty appears as a cartoon face and interacts with the students using speech, text, and an animated concept map to explain her reasoning. Students use a graphical drag and drop interface to create and modify their concept maps. When queried, Betty can provide answers, and explanations for how she derives her answers, and simultaneously speak and illustrate the derivation process on the concept map by animation. The interface of Betty's Brain is implemented in Java with Java Swing components. In the sections below, we describe the software's three modes: *TEACH*, *QUERY* and *QUIZ* that reflect generic teaching activities.

TEACH Betty: As mentioned previously, students teach Betty using a concept map interface. Figure 1 displays an example of a concept map used to teach Betty about the river ecosystem. The map shown is not complete.

QUERY Betty: Students are able to query Betty about what they have taught her. Betty has templates for two types of queries:

1. What will happen to concept A when we increase/decrease concept B?
2. Tell me what you know about concept A.

To answer questions in the form above, Betty uses a simple chaining procedure to deduce the relationship between a set of connected concepts by propagating the effects of the change and resolving loops in the concept map through a process described in Biswas et al. (2005). After Betty answers a question, the student can then ask her to explain how she derived her answer. Betty verbally explains her answer while simultaneously animating the concept map.

QUIZ Betty: During the quiz phase, students observe Betty's responses to a set of pre-scripted questions. A mentor agent informs Betty (and the student) if Betty's answers are right or wrong. The mentor also gives hints for each question to help the student correct errors in the concept map they have created. The system implements three levels of hinting, from suggesting that the student read a section of the resource materials to directly telling the student how to correct the concept or link in the concept map.

The initial phase, much like a real teaching task (preparing to teach) consists of the students learning for themselves by reading resources provided to them and attempting to organize what they are learning so they can teach Betty using the concept map representation. During this phase, there is little response from Betty but use of the map building interface is high. Therefore, it is important that the interface be easy-to-use because learners are struggling to convey domain knowledge in concept map form and are not experiencing any interaction with the agent that supports the learning activity or provides motivation. Once the learner is past the initial teaching phase, the level of interactions with the agent increase significantly through the query, explanation, and quiz mechanisms. Our studies have shown that the learners are motivated by the responsibilities of continuing to teach Betty so she can do well in her quizzes, so they continue to refine their maps, ask her more queries, and get her to take quizzes to check how well she is doing (Biswas et al. 2005; Leelawong and Biswas 2008). Betty's quiz scores, and by extension the learner's knowledge improves with time, and, in past studies, a number of students succeeded in getting Betty to get a perfect score on her quizzes in 5-6 45 minute sessions (Tan et al. 2006).

Animation for Betty's Brain

Given the importance of initial map creation and subsequent interactions with Betty in aiding and motivating student learning, we decided to make Betty a more powerful and realistic animated agent, which she was not in the prior studies. Animation allows us to explore emotive behaviors that may further engage the learner, as has been shown and associated with improved learning outcomes (Baylor and Ryu 2003; Elliott et al. 1999; Towns et al. 1998). As mentioned previously, our goal is to create an agent that is easily customizable, which may in the future allow us to explore more fully the psychometric structure of a teachable agent as has been done for a pedagogical agent (Ryu and Baylor 2005). In addition to emotional expressions, we design the teachable agent to speak and animate synchronously as she speaks. We use synthesized speech, although studies have shown that learners may be affected by how real the agent's voice is (Forbes-Riley et al. 2006; Atkinson et al. 2005).

The technical goal is to synthesize an easily realizable, controllable videorealistic character that includes a speech engine. The character should resemble a human, have lip movements synchronized with speech, and demonstrate different facial expressions that are related to their performance on the system. For example, Betty should appear happy if she does well on a quiz or her quiz scores improve. Similarly, she should have a sad expression if she has not done well, and if her quiz scores go down. There have been several systems that produce videorealistic animation

(Ezzat et al. 2002; Bregler et al. 1997; Blanz et al. 2003). These systems are quite powerful; what distinguishes ours is the speed with which a videorealistic model can be produced. We were willing to sacrifice some quality and power to achieve this speed. For example, (Ezzat et al. 2002) report that the process required to produce a speech animation module is on the order of several days in a semi-automatic manner. In contrast, our method takes only a few hours to produce a working module, with no more manual intervention than prior systems. This fast production time is desirable in various applications such as teachable and pedagogical agents, where, for example, the agent may be customized to the application as it is deployed.

Technical Approach

The key idea of this work is the use of nonlinear dimensionality reduction techniques to create a fast parameterization of the corpus of video footage, and then to play the footage back in a novel manner as directed by the circumstances of the teachable agent system, on a frame-by-frame basis. Dimensionality reduction has been a component of several animation techniques. The traditional technique is linear dimensionality reduction using Principle Component Analysis (PCA) (Jolliffe 1986), a technique that generates mean data and eigenvectors that span the principle shape variations in the data space. PCA generates a k-dimensional subspace that optimally approximates a zero-mean data set in a least squares sense. Multidimensional scaling (MDS) (Kruskal and Wish 1978) represent a set of alternative techniques that find an embedding that preserves the pairwise distances. It is equivalent to PCA when those distances are Euclidean. PCA has been used by (Guenter et al. 1998) as a component of their facial animation system, and more recently by (Safonova et al. 2004) to represent motions in a lower dimensional space for dynamic optimization.

We have found that linear techniques (PCA, classical MDS) do not provide adequate dimensionality reduction to solve the problems we discuss below. This inadequacy manifests itself by requiring a high dimension to reduce standard error heuristics to reasonable values, and interpolation or blending methods not producing acceptable results when used in these spaces. In other words, even when preserving the computational burden of high dimensionality, simple blending gave unpleasant visual artifacts. We therefore examined nonlinear techniques, which can capture the dimensionality and geometric structure of a larger class of manifolds. Kovar and Gleicher (2004) use similar ideas in their system to build new animation from large datasets. Grochow et al. (2004) do style-based posing of articulated figures using Gaussian Process models. These models, while elegant, in our experience have a tendency to get trapped in local minima and thus require expensive optimization machinery.

The technique employed in this paper for manifold-based nonlinear dimensionality reduction is Isomap (Tenenbaum et al. 2000). Locally Linear Embeddings (LLEs) (Roweis and Saul 2000) are a related technique. Both methods use local neighborhoods of nearby data to find a low-dimensional manifold embedded in a high-dimensional space. There is a large literature on nonlinear dimensionality reduction, and alternative techniques are available, e.g., (Hastie and Stuetzle 1989; Brand 2002). Isomap is a polynomial-time algorithm that reaches a global minimum for the data. An extension of the Isomap technique to incorporate temporal structure into the data was developed by (Jenkins and Matari´c 2003) and is called ST-Isomap. They focus on synthesizing humanoid motions from a motion database by automatically learning motion vocabularies.

In our work, we use Isomap to parameterize a corpus of video data, then walk the computed manifold based on the output of a text-to-speech system to synthesize speech. Isomap works by computing an embedding of the data within the original data space, giving a one-to-one correspondence between the original data and the embedded data. Given a data set living in a high-dimensional space X with distance function D_X both techniques pursue a three-step algorithm to compute the embedding, as follows:

1. Compute local neighborhoods based on the distances $D_X(i, j)$ between all pairs of points i and j in the input space X. These local neighborhoods are clusters of points for which the input space metric closely approximates the geodesic distance, e.g., the distance along the manifold. Next we define a neighborhood graph G by connecting i to j if it is one of the K nearest neighbors of j. The edge length is set to $D_X(i, j)$.
2. Globally connect the neighborhood graph G by estimating the geodesic distances between all pairs of points. This distance matrix, D_G is determined by computing the shortest paths between all pairs of points in G using Floyd's algorithm (Sedgewick 2002).
3. Embed the data into a lower dimensional space using MDS.

Isomap will find a good lower dimensional manifold as long as the data set is sufficiently dense and can form a single connected component. The computations in Isomap guarantee a global minimum, and the dimensionality of the embedding can be estimated by examining the residual variance of the output of the MDS stage of the algorithm. A key to finding a good embedding is finding a good distance metric with which to compute the local neighborhoods (Seward and Bodenheimer 2005). For this work we simply use the vector difference of two frames as our metric.

An Enhanced Interface for the Teachable Agents System

A previous study conducted in the Nashville Metropolitan school system indicated that the user interface needed improvement (Davis et al. 2003; Biswas et al. 2005). On the one hand, students attributed an independent identity to their agent, Betty, though they knew she was just a computer program because she could answer questions and explain her answers with the concept maps they taught her, sometimes better than they could (?). But it took students quite some time to learn how to use the concept mapping interface, and the query and quiz functions in an effective way, which made them less focused on learning and teaching domain knowledge. Other shortcomings that students pointed out in exit interviews were that Betty, as a good student, should be more interactive, e.g., "to react to what she was taught" or "do some sort of game or something and make it more interactive." The previous Betty character was quite passive, and only responded when the student asked her questions or asked her to take a quiz. Reflecting on the study results and student feedback, we decided that Betty needed to demonstrate more qualities of good human students in a more convincing manner. At the same time, we redesigned the interface to make it more self-explanatory so that the students found it intuitive and easy to use with minimal training.

Figure 3 shows our enhanced interface for Betty's Brain that makes the system easier to use for novices. These changes are implemented as two inter-related factors: (i) visual, which includes aesthetic appearance, look-and-feel, and the presence of a videorealistic character, and (ii) cognitive, which is directed to functionality, and information representation and interpretation of the features provided by the interface.

Causal links form an important part of the concept map structure. The original interface differentiated between the increasing and decreasing effects by using the symbols (++) and (–), respectively. This representation was confusing to our young users, therefore, we decided to use colors to more explicitly represent the semantics of the causal links: red is used to represent a decreasing effect, and green represents an increasing effect. We kept the symbols (++) and (–) along with the colors to provide redundant but very useful information. The sign information should provide help to users who are color-blind.

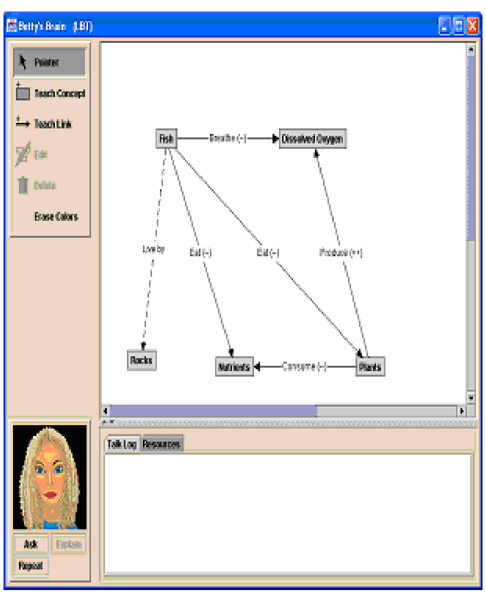

Figure 2: Original Betty's Brain Interface

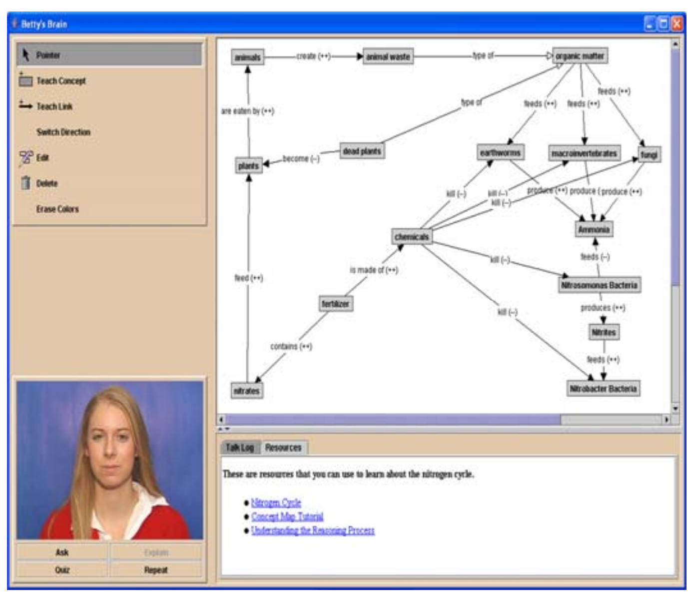

Figure 3: Enhanced Betty's Brain Interface

Data from the prior study indicated that some of the common errors students made were (i) the incorrect representation of the effect of a causal link (creating a causal link with increasing effect instead of a decreasing effect), and (ii) specifying the direction of the arrow incorrectly. When students got into the monitoring phase using the quiz or by asking queries, they found it hard to pin point these errors, and even when they did, they found it frustrating to correct them using the original interface (Davis et al. 2003; Biswas et al. 2005). For example, if students had mistakenly created a link that indicated "Earthworms produce ammonia causing ammonia to decrease" changing the relation of the link from "decrease" to "increase" involved selecting the link by clicking on it, deleting it, and then recreating a causal link of the appropriate type from "Earthworm" to "Ammonia."

In the new interface, two buttons — "Switch Trend" and "Switch Direction" — were added to help students easily rectify these common mistakes with a single click. For example, to change a trend in the relation "Animals eat plants causing plants to increase", the student would select the link and click on the "Switch Trend" button to rectify the mistake. The "Switch Direction" button changes the direction of the link selected.

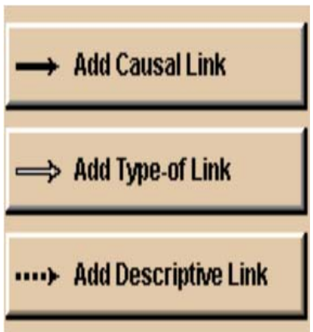

Figure 4: Original Individual Buttons

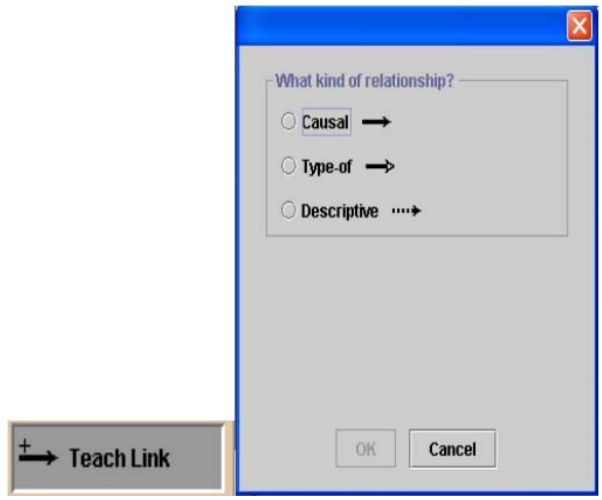

Figure 5: Pop-up window displayed in the new interface when the "Teach Link" button is clicked."

An issue in the introduction of these buttons was the concern that students might use them to employ a "quiz-edit-quiz" strategy (Wagster et al. 2007). Since students found it hard to pinpoint the error, especially for answers that involved longer chains of reasoning, making it easier to switch trends or directions would lead to students developing a strategy to "game" the system, i.e., make repeated changes to the current concept map, and after ever change take the quiz to see if previously incorrect answers were now correct. This strategy is clearly undesirable – it implies a "guess and check" approach to learning instead of learning with understanding. To guard against it, we added a pattern tracking agent to detect this pattern of activity by the students. When this is detected, Betty balks at taking a quiz, and indicates to the student that she feels that she "has not learned anything new, and, therefore, (she) may not do well on the quiz." She suggests to the student that they read the resources, teach her more, and check if she has learned by querying her before sending her to take the quiz again. This guided feedback has been shown to be more effective in promoting a positive learning experience than simply allowing the "quiz-edit-quiz" strategy (Tan et al. 2006; Wagster et al. 2007). The original interface provided three different buttons for adding a link in the concept map, corresponding to the three different types of links. Though currently screen space is not an issue, in the future when new features are added there may be space constraints. Therefore, these three buttons were replaced with just one, labeled "Teach Link", in the new interface. On clicking this button, a window pops up as shown in Figure 5. The users can select from this window which type of link they want to create.

Animated Agent Construction

Our method is to record a sequence of video with a subject speaking visemes, facial images corresponding to phonemes; in English there are 23 visemes. Additionally, the subject records several facial expressions such as a happy, sad, frustrated, and bored. These emotions represent states that we believed applied important affect: happy when performing well, sad when performing poorly, bored and frustrated when it seemed the students actions were not constructive. This sequence of video is typically about 40 seconds long. Individual frames of the 23 visemes are then manually selected (Figure 6, from left to right, the "v" sound in river, the vowel sound in "ear," and the vowel sound in "too."). This process takes about 15 minutes. To make the data set more dense, a random number of images from the recorded sequence are selected. We experimented with several values and found that 200 extra frames works well. The resulting data set is then downsampled by a factor of 8 to 90 by 60. Isomap is applied to the downsampled data set and a dimensionality for the embedding is selected. This part of the process takes about two hours. In our experience, a three-dimensional embedding with a neighborhood size of three works well. At this point we can traverse the graph, and, using the high-resolution images, create animation. The Isomap graph in two dimensions and the residual variances (error for dimensionality reduction) are shown in Figure 7. In this figure, the individual frames of video are denoted by the circles, and the graph edges represent the connections between frames that are similar. Each circle represents a frame of video, reducted from its original high dimension (5400) to two dimensions (in the Figure, three dimensions internally). This reduction in dimension comes at the cost of the error shown as the residual variance in the Figure. Thus, the error of the embedding from its original high dimension (5400) to dimensions of 10 and fewer are shown in graph on the right. Selecting the "knee" at dimension three works well for all examples we have tried. The graph of frames, hence of speech, thus constructed is then used in the animation as described below.

In our first example, the goal is to use Isomap to create a speaking, animated character whose lip movements are synchronized with speech, and whose facial expressions display some basic emotions that are linked to learning tasks. In addition to the 23 visemes, we captured facial images portraying six emotional states such as sad, angry, and confused. We segmented the image into two parts. When the character was speaking, the emotional state of the character was conveyed primarily by the character's face, neck, and shoulders. When the character was not speaking, the mouth depicted the emotional state of the character. To accomplish this two stage behavior (speaking and showing emotion), we had to do additional processing.

Although the subject was instructed only to move their lips in the training video, some translation and rotation of the head, neck, and shoulders did occur. To stitch the animation of the mouth onto a foreground image, the video frames had to be aligned so that the character was in the same position. To align these images, contour points of the subject's silhouette in each frame were extracted, and then used to register each successive frame to the first frame using an iterative closest point method (Besl and McKay 1992). The alignment process took about two hours. Once the set was aligned, the mask used for the mouth images was created by hand drawing the outline of the mouth and jaw on a single frame of the aligned images. The seams of the mouth images were blended with the foreground image. Figure 8 shows this process.

The mouth movements are controlled by a text-to-speech engine so that sound and animation are synchronized. When a word is to be spoken, our software breaks the word into visemes. Additionally, the character's expression is controlled by an emotional state variable. The emotional state variable is controlled by an algorithm that assesses several factors in the state of the learning process. As mentioned above, Betty can have an emotional state of unhappy, for example, if she does not do well on a quiz; she can be frustrated if she is not taught a sufficient amount between tests. And she is happy, when she passes a quiz.

When these states change, a transition to a new expression must be made. Both of these changes can occur simultaneously using the blending method described above. The emotional state traverses the graph on one path, while the speech part traverses the graph along another. The blending algorithm stitches the moving mouth into the emotional image of the state. Our graph traversal is done as follows. To transition between two visemes, we find the midpoint of a straight line in the embedding space that connects the two visemes. We then find the closest image to that midpoint in the embedded space. Due to the rate at which the character speaks, the transition between visemes can only be a single image and maintain synchronization with the audio (Figure 9). Nonetheless, it seems realistic. If the character's mood changes, the foreground image will transition to a new mood by approximating the straight line connecting the two images. When transitioning from emotional states, interpolation images are selected along uniformly spaced intervals of the connecting line segment. As a result of these transitions found by the Isomap algorithm, we have succeeded in creating a videorealistic animated model that can speak any word in real time based on its visemes.

Figure 6: Three video image each representing a viseme

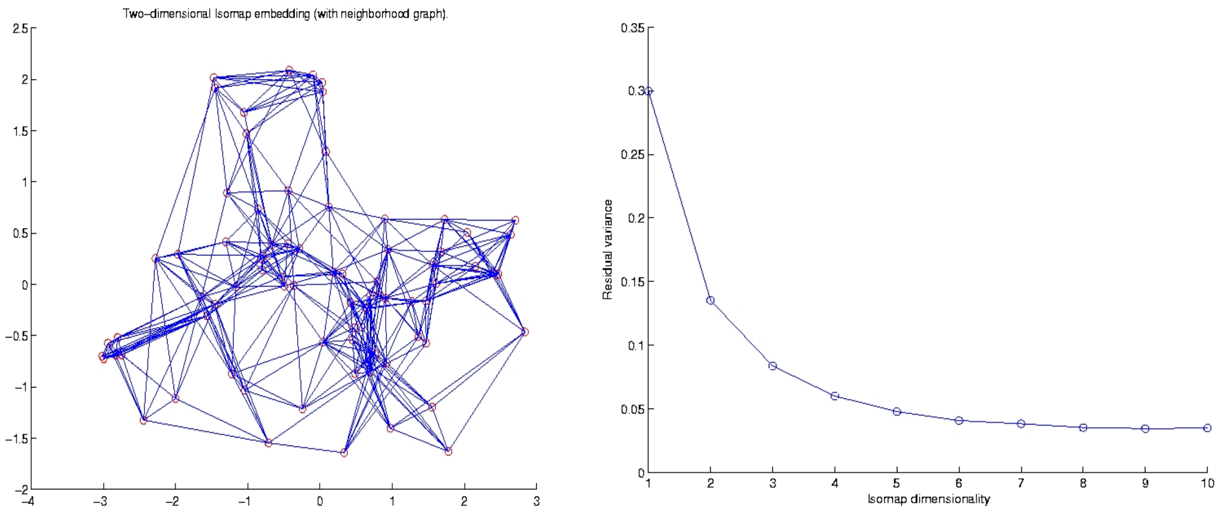

Figure 7: The Isomap embedding (left) and residual variance (right) for first example. For purposes of illustration, the figure on the left shows the graph of shortest path distances between images of facial expressions, reduced in dimension to a two-dimensional space. The residual variances shows the error of the embedding process from its high dimension to its low dimension. The “knee” of the curve is three dimensions.

Figure 8: Emotional expression with silhouette (left), masked viseme (middle), and blended image (right)

Figure 9: The middle image is the interpolating frame for the left and right images

A second character is shown in Figure 9. A subject is again recorded speaking. However, for this character, we do not align our dataset, nor do we change the emotional state of the character. We simply extract the visemes and additional frames that are used for transitions, and omit the registration steps. The result is a videorealistic character with natural head and shoulders movement. The character shown in Figure 9 is a prototype for a mentor agent and is included here for comparison. It was not used in the evaluation studies discussed next.

Evaluation

Interface Enhancements without Animation

To evaluate the effectiveness of our changes, we conducted a usability study on the interface. Study participants were 48 high achieving students aged 9-11 (U.S. fifth-grade) from an urban public school located in Metro Nashville. High achieving students were selected to avoid confounds associated with poor reading ability. The students were part of a larger study that investigated the effects of self-regulated learning using Betty's Brain, and were familiar with working on Betty's Brain by the time this study was conducted (Leelawong and Biswas 2008). Hence no training in working with the interfaces was needed, though the new features of the interfaces were shown and explained to them. The user study had three phases: introduction, testing, and questionnaire sections.

In the testing phase, the students were asked to correct a relatively simple concept map that had two errors (the complexity of the concept map is similar to that shown in Figure 2). Half the students worked on the new interface first while the other half worked on the old interface first. Then they switched the interfaces they were working on, now carrying out the tasks on a different concept map. The order of the concept maps was also switched to remove any bias that could possibly be created by the different maps used.

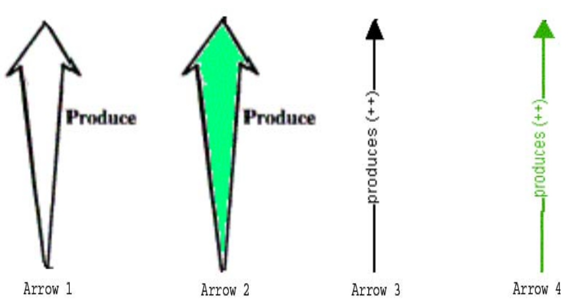

Figure 10: The four options to represent a causal link with increasing effect

	High Preference to Lowest			
Arrow 1	2	10	15	21
Arrow 2	20	16	18	4
Arrow 3	2	8	16	22
Arrow 4	24	14	9	1

Table 1: Student's feedback in the questionnaire section regarding causal links

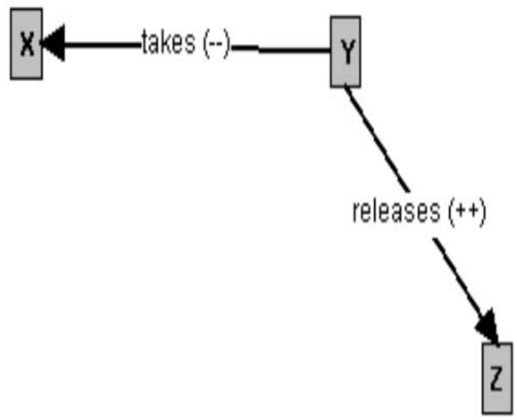

Figure 11: If Y decreases, what happens to X and Z?

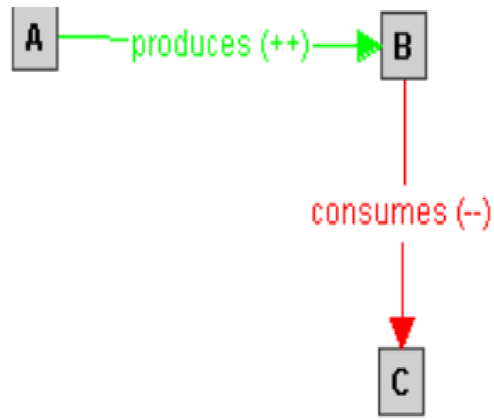

Figure 12: If A decreases, what happens to B and C?

Finally users answered a paper questionnaire. In the questionnaire, they were asked to compare features in the two interfaces. Also, to find out how color affects the understandability, the students were shown concept maps outside of the river ecosystem domain, and they were asked to answer questions on them. Part of the questionnaire asked for preferences only, and did not evaluate a learning related outcome associated with those preferences.

Discussion of Results

Log files were generated for each section of the test tracking every action of the user along with a time stamp. These timings were used for comparing the two interfaces. They were analyzed using a two-way ANOVA to control the effect of the ordering of the interfaces.

When correcting a concept map, learners were faster using the new interface (mean 66.9s) than using the old (mean 124.41s). These results are statistically significant ($F = 14.92$, $MS = 72680$, $p < .01$). These values imply that the students found it easier to go about their teaching and learning activities using the redesigned user interface. The effect of the ordering of the interfaces on their time to perform activities was not significant. This result is reasonable since, as mentioned before, the users were experienced with the software and thus the order in which the interfaces were presented to them did not have any significant effect. Neither were interaction effects significant.

In the questionnaire, the users were asked to mark their preference regarding adding the links — whether they liked to have separate buttons for each link or they liked the pop-up style. 33 out of the 48 users preferred the pop-up style, whereas the other 15 preferred to have separate buttons. This preference is statistically significant ($c2 = 6.75$, $p < .01$). Students were also asked which style of arrow (shown in Figure 10) they preferred to represent causal links for increasing and decreasing effects. They were asked to rate them according to order of preference. Table 1 shows results of this ranking for the four different options shown in Figure 10. As can be seen from this table, the users felt that the arrows used in the new interface (arrow 4) was better than the format used in the previous interface (arrow 3). The positive effect of color is clearly substantiated by their choice of arrow 2 over arrow 1.

However, other related items on the questionnaire presented a different picture. The questionnaire had two concept maps - one using normal block arrows (as in the old interface) and another, which used colored arrows (as in the new interface). The students were asked to answer two simple questions of the type "If A increases what happens to B and C ?" as shown in Figures 11 and 12. Nearly half of the students (23 out of 48) answered this question incorrectly for the colored concept map, compared to only one-fourth (12 out of 48) for the uncolored one. Thus, the number of incorrect answers to these questions was twice as high for the colored concept map as for the uncolored one. This result reveals that the students had incorrectly understood the meaning of the colored links. This flaw was not exposed when the students worked on the full system as they were working on a concept map in a domain they were familiar with. Asking them questions outside this domain helped us to identify this problem. Though the changes to the user interface made it simpler and more efficient to use, it suffered from the primary drawback of being incorrectly interpreted by the students when they were involved in reasoning tasks. In the more abstract domain for the latter questions, students used the color on the arrow as an absolute. In other words, they inferred that "red" meant “decrease” (similarly, “green” meant an “increase”) to the destination concept “B” regardless of whether “A” itself increased or decreased. Therefore, we decided not to use colored arrows in future versions of the system.

Animated Agent Evaluation

Evaluation of the animated agent was done as a separate study to the one above. This study involved 36 U.S. students aged 9-11 (fifth-grade) from the same public school as the previous study. A reasonable indicator of the effectiveness of an animated agent would be to assess task performance with and without the animation. Thus, we presented a task in which students were asked to find mistakes in a concept map and correct them. This “debugging” task involved the nitrogen cycle for plants. The concept map shown in Figure 3 had three errors: the relation between (i) “dead plants” and “plants,” (ii) “Ammonia” and “Nitrosomonas Bacteria” are incorrect, and (iii) an important relation between “Nitrobacter bacteria” and “nitrates” is missing. The control group for this study worked with the non-animated agent whereas the experimental group worked with the animated agent. Both agents used the text-to-speech synthesizer to coordinate speech and lip movements. Students were given 45 minutes, and the resulting concept maps were scored in a condition-blind manner. Our previous experiences have found students at this level find such tasks extremely difficult (Leelawong and Biswas 2008), which explains the overall low scores. However, students using the animated interface scored 1.13 out of a possible three, whereas students without the animated interface scored 0.59 out of three. These results are statistically significant ($F = 4.32$, $MS = 2.64$, $p = .045$).

Additionally, a five-point Likert scale survey and questionnaire asking students for comments on the effectiveness of the interfaces was administered. Answers were coded and evaluated using a Wilcoxon rank sum test. In response to questions about Betty’s emotions and feelings, students responded that the animated version had significantly more emotion and showed feelings ($p = .01$ in both cases). Students were marginally more motivated to teach the animated agent ($p = .07$) but were significantly more likely to want the animated agent to learn everything ($p < .05$). Students commented that they liked the agent’s facial expressions and emotions. Most of the negative comments centered around the quality of the voice of the agent that was produced by the text-to-speech synthesizer.

Discussion, Conclusions, and Future Work

This paper has presented a methodology and its implementation that has led to an improved design for the user interface as well as the animated teachable agent in our Betty’s Brain system. Two experimental studies, one that compared the new interface to the existing one, and the second that compared the animated agent with facial expressions that demonstrated some emotion with the unanimated agent have demonstrated positive results with U.S. “middle”-schools, aged 9-11. (U.S. middle schools typically encompass ages 9 through 14, but the domain of knowledge for our particular study is taught to students aged 9-11). As educational technology, our focus on design of teachable agent systems has been to create useful social interactions between virtual agents and students that promote learning in exploratory and constructivist learning environments (e.g., (Bandura 1989; Palinscar 1998)). This approach is different from some traditional tutoring systems (Wenger 1987), where the focus in on immediate correction of student errors during problem solving tasks. Though exploratory environments have been hypothesized to provide better learning opportunities for students, unless they are well-designed, intuitive, easy to use, and provide the right kind of feedback, they may often frustrate students, when they get stuck addressing difficult problems that may involve learning new knowledge as well as monitoring their learning processes (Lajoie and Derry 1993). We have been studying all of these issues in the context of our teachable agents system (Schwartz et al. 2007). While cognitive load per se was not directly measured, we believe we can reasonably infer that our redesigned user interface and the use of more natural animation for the teachable agent has helped in reducing the cognitive load that students face when teaching by creating causal concept maps, and when they monitor these maps to find and correct errors in their initial maps. This inference is reasonable because of performance and learning improvements made by the students in the absence of any other changes to the system. More specifically, our studies show that students preferred the pop-up mechanism for adding the links. This feature saves valuable screen space, which has been used in subsequent versions of the system for providing other helpful learning tools (For a newer version of the system go to http://www.teachableagents.org/bettysbrain). The introduction of specialized buttons for “Switch Trend” and “Switch Direction” made it easier for students to make corrections to the causal relations they were specifying in their concept maps. Of course, like many other features, this set produced a “double-edged sword.” Easier modifications helped students focus more on the monitoring task, but it also facilitated the use of a suboptimal “guess and check” strategy. Once we discovered this, a pattern tracking agent was inserted that discouraged and sometimes prevented students from engaging in this behavior.

Another feature that we experimented with was the use of color to explicate the increase/decrease relations in the concept map. Earlier experiments conducted with college-age students (Viswanath et al. 2004) reported improved efficiency in using the system and the use of colors did not interfere with their understanding of the reasoning mechanism. In the present study, middle school students were more efficient with the colored links when they created their own maps, and in the questionnaire they indicated their preference for the colored links. However, when given debugging tasks with others’ maps, the students misinterpreted the meaning of color on the links. One interpretation of this in computer science and cognitive terminology may be that the use of color was “overloaded”, and the middle school students were not mature enough learners to handle this overloading. In subsequent implementations of Betty’s Brain, we do not associate colors with links. Instead, when Betty reasons, the borders of the concept boxes are colored green or red to indicate an increase or decrease in the particular entity. We have found that this coloring scheme helps students understand the reasoning mechanism, and it does not create the confusion associated with the colored links.

The use of the more videorealistic face with a few animated expressions has also been shown to be helpful, primarily because the students seem to be more motivated to create, monitor, and correct their concept maps. This was clear from the improved performance on the debugging tasks and the students responses to questions after they had finished their tasks. Therefore, animation is useful in the teachable agent environment, primarily because it addresses the issues of social learning and motivation that we discussed earlier. However, there is a trade-off because authoring animated agents can be expensive and time consuming. The machinery that we have developed as part of this research allows quick and reasonably believable animations for customizing different agents that we may employ in the teachable agent system. One of our future goals is to provide students with more choice, so we can easily support different learning style preferences by adjusting the personality and appearance of the agent. We can also easily change the expression and appearance of the agent to meet specifications for different domains of knowledge. These results on the utility of animation are consistent with findings for pedagogical agents (Baylor and Ryu 2003). Investigating different rendering styles for agents, such as non-photorealistic rendering (Gooch and Gooch 2001) and full body animation (Rose et al. 1998) using this machinery is a topic of future work. Allowing full body animation will allow us to incorporate gestures into the agent as well.

An important area of current research is how much scaffolding for learning the interface an animated agent can supply when coupled with enhanced content and cognitive feedback as described by Tan et al. (2006). Future experiments will evaluate how interface modifications affect learning-related outcomes. For example, many simple changes may create an interface that is sufficiently easy to use but will allow users to exploit shortcomings in the reasoning portions of the system with the result that they are able to “game” the system, i.e., have Betty achieve good results without the students themselves learning much. Also, future work in this field will explore the judicious use of colors and shapes to represent hierarchies and relationships between entities. The use of these in animations to explain the complex relations in a system will also be investigated.

References

Armis, L. F. (1983). The process and effects of peer tutoring. *Human Learning, 2*, 39–47.

Atkinson, R. K., Mayer, R. E., & Merrill, M. M. (2005). Fostering social agency in multimedia learning: examining the impact of an animated agent’s voice. *Contemporary Educational Psychology, 30*, 117–139.

Bandura, A. (1989). Social cognitive theory. In R. Vasta (Ed.) *Annals of Child Development,. 6, Six theories of child development.* Greenwich, CT: JAI Press, 1-60.

Bargh, J. A., & Schul, Y. (1980). On the cognitive benefits of teaching. *Journal of Educational Psychology, 72*, 593–604.

Baylor, A. L., & Ryu, J. (2003). Effects of image and animation on agent persona. *Journal of Educational Computing Research, 28(4)*, 373-395.

Baylor, A. L., Shen, E., Warren, D., & Park, S. (2004). Supporting learners with math anxiety: The impact of

pedagogical agent emotional and motivational support. Presented at *Social and Emotional Intelligence in Learning Environments Workshop*, International Conference on Intelligent Tutoring Systems, Maceió, Brazil.

Baylor, A. L. (2005). The impact of pedagogical agent image on affective outcomes. In C. Conati, S. Marsella, & A. Paiva (Eds.) *Affective Interactions: The Computer in the Affective Loop,* Workshop at the *International Conference on Intelligent User Interfaces*, San Diego, CA.

Besl, P. J., & McKay, N. D. (1992). A method for registration of 3-d shapes*. IEEE Transactions on Pattern Analysis and Machine Intelligence,* 14, 119–128.

Biswas, G., Schwartz, D., & Bransford, J. (2001). Technology support for complex problem solving: From SAD environments to AI. In K Forbus & P. Feltovich (Eds.) *Smart Machines in Education: The Coming Revolution in Educational Technology*, Cambridge, MA: MIT Press, 71-97.

Biswas, G., Schwartz, D., Leelawong, K., Vye, N., & Tag, V. (2005). Learning by teaching: A new agent paradigm for educational software. *Applied Artificial Intelligence 19*(3), 363–392.

Blanz, V., Basso, C., Poggio, T., & Vetter, T. (2003). Reanimating faces in images and video. *Computer Graphics Forum 22*(3), 641–650.

Brand, M. (2002). Charting a manifold. *Neural Information Processing Systems*, *15*, 961-968.

Bregler, C., Covell, M., & Slaney, M. (1997). Video rewrite: Driving visual speech with audio. *Proceedings of SIGGRAPH 97*. In *Computer Graphics* Proceedings, Annual Conference Series, ACM SIGGRAPH, Los Angeles, CA, 353–360.

Chi, M. T. H., Siler, S. A., Jeong, H., Yamauchi, T., & Hausmann, R. G. (2001). Learning from human tutoring*. Cognitive Science 25*(4), 471–533.

Cole, R., Massaro, D. W., de Villiers, J., Rundle, B., Shobaki, K., Wouters, J., Cohen, M., Beskow, J., Stone, P., Conners, P., Tarachow, A., & Solcher, D. (1999). New tools for interactive speech and language training: Using animated conversational agents in the classrooms of profoundly deaf children. In *Proceedings of ESCA/SOCRATES Workshop on Method and Tool Innovations for Speech Science Education,* London, UK, 45-52*.*

Craig, S. D., Gholson, B., & Driscoll, D. (2002). Animated pedagogical agents in multimedia educational environments. *Journal of Educational Psychology 94*, 428–434.

Davis, J., Leelawong, K., Belynne, K., Bodenheimer, B., Biswas, G., Vye, N., & Bransford, J. (2003). Intelligent user interface design for teachable agent systems. In W. L. Johnson, E. Andre, & J. Domingue (Eds.) *2003 International Conference on Intelligent User Interfaces*, Miami, FL, 26–33.

Elliott, C., Rickel, J., & Lester, J. (1999). Lifelike pedagogical agents and affective computing: An exploratory synthesis*. Lecture Notes in Computer Science*, *1600*, 195–212.

Ezzat, T., Geiger, G., & Poggio, T. (2002). Trainable videorealistic speech animation. *ACM Transactions on Graphics 21*(3), 388–398.

Forbes-Riley, K., Litman, D., Silliman, S., & Tetreault, J. (2006). Comparing synthesized versus pre-recorded tutor speech in an intelligent tutoring spoken dialogue system. In G. Sutcliffe & R. Goebel (Eds.), *Proceedings of the 19th International Florida Artificial Intelligence Research Society Conference*, Menlo Park, CA: AAAI Press, 509-514.

Gooch, B., & Gooch, A. (2001). *Non-Photorealistic Rendering*. Wellesley, MA: A. K. Peters.

Graesser, A. C., Person, N., & Magliano, J. (1995). Collaborative dialog patterns in naturalistic one-on-one tutoring. *Applied Cognitive Psychologist 9*, 359–387.

Grochow, K., Martin, S. L., Hertzmann, A., & Popovic, Z. (2004). Style-based inverse kinematics. *ACM Transactions on Graphics 23*(3), 522–531.

Guenter, B., Grimm, C., Wood, D., Malvar, H., & Pighin, F. (1998). Making faces. *Proceedings of SIGGRAPH 98.* In *Computer Graphics* Proceedings, Annual Conference Series, ACM SIGGRAPH, Orlando, FL 55–66.

Hastie, T., & Stuetzle, W. (1989). Principal curves. *Journal of the American Statistical Association, 84*, 502–516.

Jenkins, O. C., & Mataric, M. J. (2003). Automated derivation of behavior vocabularies for autonomous humanoid motion. In *Proceedings. of the 2nd international joint conference on autonomous agents and multiagent systems*, Melbourne, Australia, 225–232.

Jolliffe, I. (1986). *Principal Component Analysis*. New York: Springer-Verlag.

Kinchin, I. M., Hay, D. B., & Adams, A. (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.

Kover, L., & Gleicher, M. (2004). Automated extraction and parameterization of motions in large data sets. *ACM Transactions on Graphics 23*(3), 559–568.

Kruskal, J. B., & Wish, M. (1978). *Multidimensional Scaling*. Beverly Hills, CA: Sage Publications.

Lajoie, S., & Derry, S. (1993). *Computers as Cognitive Tools*. Hillsdale, NJ: Lawrence Erlbaum.

Leelawong, K., & Biswas, G. (2008). Designing learning by teaching agents: The betty's brain system. *International Journal of Artificial Intelligence in Education 18*, 3.

Leelawong, K., Wang, Y., Biswas, G., Vye, N., & Bransford, J. (2001). Qualitative reasoning techniques to support learning by teaching: the teachable agents project. In G. Biswas (Ed.) *Proceedings of the Fifteenth International Workshop on Qualitative Reasoning*, San Antonio, TX, 73–81.

Lester, J., Voerman, J., Towns, S., & Callaway, C., (1999). Deictic believability: Coordinated gesture, locomotion and speech in lifelike pedagogical agents. *Applied Artificial Intelligence, 13(4/5),* 383-414.

Michie, D., Paterson, A., & Hayes-Michie, J. (1989). Learning by teaching. In H. Jaakola & S. Linnainmaa (Eds.), *Proceedings of the 2nd Scandinavian Conference on Artificial Intelligence*, Amsterdam, The Netherlands, 307–331.

Moreno, R., & Mayer, R. E. (2000). Engaging students in active learning: the case for personalized multimedia messages. *Journal of Educational Psychology 92(4)*, 724–733.

Nichols, D. M. (1994). Issues in designing learning by teaching systems. In P. Brusilovsky, S. Dikareva, J. Greer, & V. Petrushkin (Eds.), *Proceedings of the East-West International Conference on Computer Technologies in Education (EW-ED'94), 1*, Crimea, Ukraine, 176–181.

Novak, J. D. (1996). Concept mapping: A tool for improving science teaching and learning. In D. Treagust, R. Duit, & B. Fraser (Eds.) *Improving Teaching and Learning in Science and Mathematics*. New York and London: Teachers College Press, 32–43.

Novak, J. D. (1998). *Learning, Creating, and Using Knowledge: Concept Maps as Facilitative Tools in Schools and Cooperations*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Palinscar, A. S. (1998). Social constructivist theories of teaching and learning. *Annual Reviews in Psychology 49*, 345–375.

Ramirez-Uresti, J. A., & du Boulay, B. (2004). Expertise, motivation, and teaching in learning by teaching systems. *International Journal of Artificial Intelligence in Education 14,* 67–106.

Rose, C., Cohen, M., & Bodenheimer, B. (1998). Verbs and adverbs: multidimensional motion interpolation. *IEEE Computer Graphics and Applications 18*(5), 32–40.

Roweis, S., & Saul, L. (2000). Nonlinear dimensionality reduction by locally linear embedding. *Science 290*, 2323–2326.

Russell, S. J., & Norvig, P. (1995). *Artificial Intelligence: A Modern Approach*. Upper Saddle River, NJ: Prentice-Hall.

Ryu, J., & Baylor, A. L. (2005). The psychometric structure of pedagogical agent persona. *Technology, Instruction, Cognition, and Learning, 2*, 291–319.

Safonova, A., Hodgins, J. K., & Pollard, N. S. (2004). Synthesizing physically realistic human motion in lowdimensional, behavior-specific spaces. *ACM Transactions on Graphics 23(3),* 514–521.

Schwartz, D., Blair, K. P., Biswas, G., & Leelawong, K. (2007). Animations of thought: Interactivity in the teachable agents paradigm. In R. Lowe & W. Schnotz (Eds.) *Learning with Animation: Research and Implications for Design*, London, UK: Cambrige University Press, 114–140.

Sedgewick, R. (2002). *Algorithms in C: Part 5 Graph Algorithms*, 3 Ed., Reading, MA: Addison-Wesley.

Seward, A. E., & Bodenheimer, B. (2005). Using nonlinear dimensionality reduction in 3d figure animation. In V. Clincy (Ed.) *Proceedings of the 43rd ACM Southeast Regional Conference*, *2*, 388–393.

Spiro, R. J., & Jehng, J. C. (1990). Cognitive flexibility and hypertext: theory and technology for the nonlinear and multidimensional traversal of complex subject matter. In D. Nix & R. Spiro (Eds.) *Cognition, Education, and Multimedia: Exploring Ideas in High Technology*, Hillsdale, NJ: Lawrence Erlbaum Associates, 163-205.

Stoyanov, S., & Kommers, P. (1999). Agent-support for problem solving through concept mapping. *Journal of Interactive Learning Research 10*(3), 401–425.

Tan, J., Biswas, G., & Schwartz, D. L. (2006). Feedback for metacognitive support in learning by teaching environments. In R. Sun (Ed.), *Proceeding*s of *28th Annual Cognitive Science Society,* Austin, TX: Cognitive Science Society, 828-833.

Tenenbaum, J. B., de Silva, V., & Langford, J. C. (2000). A global geometric framework for nonlinear dimensionality reduction. *Science 290*, 2319–2323.

Towns, S. G., FitzGerald, P. J., & Lester, J. C. (1998). Visual emotive communication in lifelike pedagogical agents. In *Proceedings of the Fourth International Conference on Intelligent Tutoring Systems*, *Lecture Notes in Computer Science*, *1452*, 474–483.

Viswanath, K., Balachandran, R., Kramer, M. R., & Bodenheimer, B. (2004). Interface design issues for

teachable agent systems. In L. Cantoni & C. McLoughlin (Eds.), *Proceedings of World Conference on Educational Multimedia, Hypermedia, and Telecommunications 2004*, Chesapeake, VA: AACE, 4197–4204.

Wagster, J., Tan, J., Wu, Y., Biswas, G., & Schwartz, D. (2007). Do learning by teaching environments with metacognitive support help students develop better learning behaviors. In D. S. McNamara & J. G. Trafton (Eds.), *Proceeding*s of *29th Annual Cognitive Science Society,* Austin, TX: Cognitive Science Society, 695–700.

Wenger, E. (1987). *Artificial Intelligence and Tutoring Systems*. San Francisco, CA: Morgan Kaufmann.