AC 2011-2488: USE OF SOFTWARE AGENT-MONITORED TUTORIALS TO GUIDE STUDENT LEARNING IN COMPUTER-AIDED DESIGN, ANALYSIS AND MATHEMATICS PROJECTS

Jack L. Beuth, Carnegie Mellon University

Jack Beuth received his Ph.D. in Engineering Sciences from Harvard in 1992. He has been on the faculty of the Carnegie Mellon University Department of Mechanical Engineering since 1992. Dr. Beuth’s research is in the areas of solid mechanics and manufacturing. He is currently working on projects in four areas: The Mechanics of Laser- and Electron Beam-Based Manufacturing Processes, Interfacial Toughness Testing of Thermal Barrier Coatings, Development of an In-situ On-chip Tensile Testing Platform, and the Modeling of Fracture-Resistant Anti-Corrosion Coatings for Steels. Dr. Beuth’s initiatives in education have included the integration of computer-aided engineering projects throughout the CMU ME undergraduate curriculum. His latest research is in collaboration with the CMU Human-Computer Interaction Institute, developing software agent-monitored collaborative projects for undergraduate courses. Dr. Beuth was a recipient of the 1998 Ralph R. Teetor Educational Award. In 2000, he was awarded George Tallman and Florence Barrett Ladd Development Professorship in Mechanical Engineering. In 2005 Dr. Beuth was co-recipient of the ASME Curriculum Innovation Award. In 2009 Dr. Beuth received the Benjamin Richard Teare Teaching Award from the Carnegie Mellon Engineering College.

Rohit Kumar, Carnegie Mellon University

Rohit Kumar is a PhD Candidate at the Language Technologies Institute at Carnegie Mellon University. He received his Masters in Language Technologies at Carnegie Mellon University in 2007 and Bachelor in Engineering from Punjab Engineering College, Chandigarh in 2003. Rohit was a Research Scientist at Language Technologies Research Center, IIIT Hyderabad from 2003 to 2005. His research interests include Conversational Agents and their applications to variety of complex interactive situations.

Carolyn Penstein Rose, Carnegie Mellon University

Carolyn Rose is an Assistant Professor with a joint appointment between the Language Technologies Institute and the Human-Computer Interaction Institute at Carnegie Mellon University. She serves as an Executive Committee member of the Pittsburgh Science of Learning Center, Treasurer of the International Society of the Learning Sciences, and editorial board member of the International Journal of Computer Supported Collaborative Learning, the Journal of Educational Data Mining, and the International Journal of Human-Computer Studies. Her research is in the area of Computer Supported Collaborative Learning and Automated Discourse Analysis.

©American Society for Engineering Education, 2011
Use of Software Agent-Monitored Tutorials to Guide Student Learning in Computer-Aided Design, Analysis and Mathematics Projects

Abstract
Internet chat-based tutorials are being developed for integrating computer modeling and design and mathematics skills into mechanical engineering undergraduate and middle school outreach programs. In modeling and design projects, tutorials help students navigate complicated software interfaces while teaching fundamental concepts through dynamic dialogues between tutorial agents and student user groups. In a typical assignment, students are asked to perform a design or modeling task that includes the use of software such as a commercial finite element code or specially designed educational software. Students work in teams, but team members are distributed within a room or between remote sites, linked by a text interface. As students collaborate electronically, an intelligent agent monitors their interactions and interjects questions or comments in response to the use of key phrases, or due to other triggers. This platform is also being adapted to the collaborative teaching of mathematics skills in engineering applications. In all projects, agent-monitored tutorials are being used to help automate collaborative learning experiences and to study how students can effectively interact with each other and with the software agents. In undergraduate projects, fundamental knowledge and intuition in interpreting results are emphasized. In outreach efforts, participants are led to consider how their work relates to the broad mechanical engineering profession.

Introduction
Over the past decade, the landscape in which engineering is practiced has been radically altered by two trends. First, there have been far-reaching technological advances in computing, information-sharing, and automated manufacturing. Second, there have been increasingly strong economic and market pressures for shorter product development cycles and internationally distributed team-based product design and manufacturing. Many of the same technological advances that have transformed engineering practice also have the potential to transform engineering education. For instance, it is widely recognized that guided use of sophisticated simulation software can enable exploratory and inquiry-based modes of learning. If these capabilities are exploited effectively, then integration of simulation-based projects can efficiently increase student learning and understanding.

In a related way, internet-based instruction coupled with on-line student collaboration, and the real-time transfer of numerical product designs, numerical simulations, images and product prototypes offers a wide range of possibilities for student learning. Instructors are no longer limited to the use of traditional, on-campus, classroom-based lectures. Recognizing this, we have been engaged in ongoing research both to transform freshman level engineering instruction as well as sophomore level thermodynamics instruction using collaborative project-based learning modules. These modules not only exploit on-line collaboration capabilities, but also employ the use of programmed intelligent software agents, which monitor student interactions and then interject comments and questions designed to teach students directly and/or increase student collaborative learning by teaching each other. Early results have showed improvements in learning of over a letter grade in magnitude for students
working with a human partner and a computer agent in comparison with students working alone without the agent. In a series of controlled classroom studies we have continued to improve learning effectiveness through changes in the computer agent design. Important innovations include offering students control over the timing of feedback, using social strategies motivated by the field of collaborative group work, and developing agents that demonstrate alignment with student goals.

The underlying thesis of this research is that offering a dynamic self-paced learning environment for student use outside of the lecture room is the best practical means for integrating sophisticated design and analysis experiences into undergraduate engineering curricula. Furthermore, the machine-monitored internet chat-based tutorial environment we use to achieve this goal offers an excellent opportunity for automating and invigorating K-12 outreach efforts and for tying them naturally to more sophisticated undergraduate-level instruction.

The foundation of our approach consists of two pillars: 1) self-paced web tutorials guiding students through software use and 2) dynamic, dialogue-based tutorial interfaces which engage students in interpreting simulation results they create. The use of self-paced web tutorials as a means of efficiently integrating complex software package use into undergraduate curricula has been the subject of a long-term effort at Carnegie Mellon. The integration of an agent-monitored dialogue-based interface into software instruction represents a substantial enhancement to this approach.

As we deploy dialogue-based tutorials, we are not simply using them to enhance software instruction. We are also studying how students learn and teach each other in a tutorial-guided dynamic chat environment involving students in one or multiple groups. Our initial focus has been a first-year introductory and second-year thermodynamics mechanical engineering courses; however, our goal is to establish a template for a new approach to design and analysis instruction throughout our curriculum. Furthermore, we are also applying our agent-monitored dialogue-based tutorial platform to the task of middle school student outreach.

**Tutor Architecture and Interface**

In this research, we are adapting a prototype architecture we have developed for supporting student interactions in a broad range of activities. This unique, automated, collaborative learning platform naturally exploits chat room-style communications that are ubiquitous on the internet, and also students’ comfort and curiosity with that environment. In our prior work using a similar environment for middle school math instruction, students found the collaborative problem solving environment highly engaging. Some students commented that the collaborative environment was “way more fun” than their typical computer lab activities, and that they were disappointed when the 45 minutes lab session was over.

The idea behind its design is for a filter to process the text from an ongoing discussion as it is happening, and to build an internal model of how the conversation is progressing. Using this model, it is possible to determine where the most strategic opportunities for supporting learning exist. Figure 1 shows an overview of the architecture used to develop our prototype infrastructure. This architecture is meant to allow context-sensitive support for collaborative learning and reflection not only to be triggered based on what is happening in the discussion, but
for it to do so with awareness of how it is affecting the state of the conversation through its continuous monitoring. Thus, if an intervention is triggered erroneously and ends up having a negative effect on the collaboration, we can detect and correct that. In this way, we minimize the risk of misdiagnosing the state of the collaboration.

As displayed in Figure 1, all interface events resulting from student contributions to the chat interface and to a shared problem solving or simulation space are sent to the Filters module. Its purpose is to identify significant events in this stream that it then reflects back to the interfaces of the students. It also uses these identified events to update its internal state. Other triggers such as timers that keep track of time elapsed since the beginning of the session or since the last significant contribution of each student are also used to manipulate the Filter module’s internal state. The internal state then is used to select strategies for selecting dialogue agents to participate in the chat session in order to offer support in the form of interactive directed lines of reasoning.

In our prior experiments we have used different kinds of triggers including topic-based filters, time-outs, interface actions, and conversational actions that are indicative of the degree of engagement of the students in the discussion. Our generic architecture is meant to be easily extended to work with other types of triggers such as cues from other modalities like speech, hand sketches, etc. We continue to improve the architecture to provide richer communication and modularization. Conversational agents meant to offer support in the midst of collaborative learning interactions can be authored with the TuTalk dialogue agent authoring system\textsuperscript{13, 14}. As displayed in Figure 1, when the Filters module sends a notification to the Conversational Agents module to trigger a particular cognitive support agent, the scheduled TuTalk agent is appended to a queue of TuTalk Agents, which are then launched in turn at appropriate breaks in problem solving behavior.

![Figure 1: Architecture of the collaborative problem-solving interface with conversational agents](image-url)
Figure 2 illustrates a typical interface used for instruction, taken from a study in sophomore-level thermodynamics. As displayed in the figure, the interface contains two panels. The rightmost panel is a chat interface, which allows students to interact with each other as well as with the conversational agents that are triggered at different occasions during the problem solving session. In our outreach efforts, our collaboration interface also includes a sketching window, also with pen strokes by any team member viewable by the others. Currently, however, our monitoring of team member interactions is limited to text interactions.

The panel on the left is a problem-solving interface that allows students to collaboratively work on a given problem. The problem solving interface in the left panel was built using CyclePad and the Cognitive Tutor Authoring Tools (CTAT). The structured problem solving CTAT panel has a problem layout and a hint button. The problem layout can either be implemented using CTAT itself, or with a simulation environment such as CyclePad as currently displayed. The integration between CTAT and the problem solving interface allows direct feedback on problem solving to be provided to students. The hint messages that are provided by CTAT are displayed in the Chat Interface. Both panels of the interface maintain a common state across the participants at all times so that student group members are independently able to manipulate all of its interface elements. Actions performed by a student in either of the panels are immediately communicated and reflected on the interface of other students in the same group.

Tutor Applications
Our tutor architecture has thus far been deployed in three classroom environments: a first-year undergraduate course Fundamentals of Mechanical Engineering, a middle-school level outreach
workshop, and a sophomore-level thermodynamics course. Work is underway to further apply our methods to reinforce mathematics fundamentals within the engineering curriculum.

**Freshman Fundamentals of Mechanical Engineering**

This course is a first-year introductory course offered to engineering students potentially interested in mechanical engineering as a major. Carnegie Mellon engineering students are required to take two discipline-specific introductory engineering courses during their freshman year. The mechanical engineering course combines an introduction to fundamentals in the areas of statics, stress analysis, dynamics, fluid mechanics, and thermodynamics with activities designed to give students a broad sense of the profession. An important project within that course is an introductory computer-aided engineering (CAE) project.

In the CAE project, students use Pro/ENGINEER, Pro/MECHANICA and Pro/MANUFACTURING software from PTC to design, analyze and plan the manufacture of a simple two-headed wrench (Fig. 3). A wrench has been chosen as the focus of this project due to its simple extruded 2-D geometry and the acquaintance nearly all students have with its use. Student wrench designs are fabricated on a CNC milling machine based on the student-generated planning file. At total of 100-130 students complete this project each semester.

![Figure 3: Screenshots and activities taking place in the first-year CAE project.](image)

An important purpose of this project is to demonstrate to students the basics of computer-aided design, analysis and manufacturing tasks routinely performed by mechanical engineers. To further reinforce fundamental concepts of stress analysis and design, an agent-monitored collaborative dialogue module was added to the analysis step of the project. In that activity, students are asked to consider changes in the wrench design, and their impact on the reliability and effectiveness of the wrench. This activity builds upon moment and bending stress concepts covered in the course lecture.
Table 1: Example tutor/student interaction from the freshman course study.

<table>
<thead>
<tr>
<th>Tutor</th>
<th>Lets talk about making the wrench easy to use.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tutor</td>
<td>Are you ready?</td>
</tr>
<tr>
<td>Student 19</td>
<td>o yeah</td>
</tr>
<tr>
<td>Tutor</td>
<td>Intuitively, if you wanted to make the wrench easier to use, would you make it longer or shorter?</td>
</tr>
<tr>
<td>Student 5</td>
<td>Longer</td>
</tr>
<tr>
<td>Tutor</td>
<td>That's right. A longer wrench is better.</td>
</tr>
<tr>
<td>Tutor</td>
<td>Why is a longer wrench easier to use though! Let's look at the concept of Moment.</td>
</tr>
<tr>
<td>Tutor</td>
<td>When you use a wrench to turn a bolt, do you want a higher or lower moment?</td>
</tr>
<tr>
<td>Student 16</td>
<td>Higher</td>
</tr>
<tr>
<td>Student 13</td>
<td>Higher</td>
</tr>
<tr>
<td>Tutor</td>
<td>That's right. A higher moment means the Wrench turns easier.</td>
</tr>
<tr>
<td>Tutor</td>
<td>We know that moment = force x length. So...</td>
</tr>
</tbody>
</table>

Table 1 provides an example of an actual student/tutor interaction occurring during this activity. As the text entries indicate, in this case the tutor is very explicitly leading the student team through the design activity, asking questions and providing feedback on student answers. This level of support has been found to be appropriate for this and similar projects, leading to significant learning gains and increases in conceptual depth in discussions between students. Experiments have also been carried out with tutor interactions that are more “social” than those shown in Table 1. For instance, the tutor can show enthusiasm when students answer questions correctly, or it can detect which team members are not interacting as much as others, and encourage them (by name) to participate more fully in the discussion. Results from some of our evaluation studies are discussed below.

Middle School Outreach

A simplified version of the freshman project has also been effectively used as a 3-hour middle school outreach workshop offered on weekends. This workshop is offered as part of the Carnegie Mellon Institute for Talented Elementary and Secondary Students (C-MITES) program, targeting intellectually curious K-9th grade students with engaging activities in the arts, humanities, science and engineering.

In this workshop, students are guided step-by-step through the CAD modeling of a wrench, after an introduction to the broad subject of computer-aided engineering and the tasks of design, analysis and manufacturing (Fig. 4). Students are then given a tour of the mechanical engineering machine shop, with demonstrations of rapid prototyping and CNC machining of a wrench design (Fig. 5). In the final activity of the workshop, students are assigned to distributed groups within the classroom and asked to use an agent-monitored texting and sketching interface to collaboratively develop ideas for improving a wrench design. Students are encouraged to think broadly about what will make the wrench better.
During the collaborative design activity, student text interactions are monitored and hints are given to encourage students to think about all aspects of making a reliable, easy-to-use, and cost-effective wrench. The workshop concludes with a discussion of student group suggestions and a formal grouping of ideas into categories of 1) mechanical, 2) ergonomic and 3) cost improvements. Tutor interactions are used to lead students toward these three concepts. The ultimate conclusion of the workshop is that the task of an engineer is not to design the best product possible. It is to design a product that meets or exceeds customer expectations of reliability, is easy to use, and has the possible lowest cost. This task is difficult, but that is why engineers are paid well for their work and skills.

**Sophomore-Level Thermodynamics**

The third application of this technology and approach to teaching pursued to date is in a sophomore-level thermodynamics course. In a project for that course, students study the efficiency of thermodynamic cycles using CyclePad. CyclePad is a package developed at Northwestern University that allows students to construct and analyze a wide variety of thermodynamic cycles. As noted in the discussion of Fig. 2, because of its open architecture,
CyclePad enables monitoring of not only student text interactions but also student inputs into the software and software outputs. This enables more detailed monitoring and guidance of student group work.

Table 2: Example CycleTalk/student interaction from the sophomore thermodynamics study.

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Student 1</td>
<td>so i think the best thing to go with first is to increase the pressure entering the connector</td>
</tr>
<tr>
<td>2</td>
<td>Tutor</td>
<td>You might want to review what you saw earlier in the sensitivity analysis investigating the effect of changing P-max on steam quality and efficiency, from the top half of page 13 in your book.</td>
</tr>
<tr>
<td>3</td>
<td>Tutor</td>
<td>type: HELP WITH CHANGEPMAX if you want to discuss it with me.</td>
</tr>
<tr>
<td>4</td>
<td>Student 1</td>
<td>help with changepmax</td>
</tr>
<tr>
<td>5</td>
<td>Tutor</td>
<td>Remember the sensitivity analysis you have completed where you manipulated P-Max and observed it's effect on the efficiency of the cycle. You can refer to the figure that I have pasted onto your whiteboard for your convenience. Having seen how changing P-Max, the Maximum Pressure at the Heater Inlet, effects efficiency and Steam, do you think as a general principle the Maximum pressure should be increased ?</td>
</tr>
<tr>
<td>6</td>
<td>Student 1</td>
<td>Yes</td>
</tr>
<tr>
<td>7</td>
<td>Tutor</td>
<td>So the Efficiency of a Cycle increases by increasing the maximum pressure of the cycle. In addition to the effect on efficiency, what happens to the steam quality at the turbine exit if you increase the maximum pressure of the cycle ?</td>
</tr>
<tr>
<td>8</td>
<td>Student 1</td>
<td>Decreases</td>
</tr>
<tr>
<td>9</td>
<td>Tutor</td>
<td>Steam quality should be maintained above 0.85 to avoid any damage to turbines blades during operation. So we need to balance these concerns by increasing the maximum pressure, but not enough that the quality of the steam drops below the minimum value (0.85). Are there other limitations on increasing the maximum pressure at the heater inlet ?</td>
</tr>
<tr>
<td>10</td>
<td>Student 1</td>
<td>no?</td>
</tr>
<tr>
<td>11</td>
<td>Tutor</td>
<td>Can the devices and pipes in the cycle stand if we keep on increasing the pressure ?</td>
</tr>
<tr>
<td>12</td>
<td>Student 1</td>
<td>20000kPa</td>
</tr>
<tr>
<td>13</td>
<td>Tutor</td>
<td>Beyond a particular pressure, the pipes and devices in a cycle become unsafe at risk for accident. The maximum pressure at heater inlet should be within safety constraints of the material used.</td>
</tr>
<tr>
<td>14</td>
<td>Student 2</td>
<td>so to what value do you want to change the pressure at s1?</td>
</tr>
<tr>
<td>15</td>
<td>Student 2</td>
<td>do we want to keep X of s3 above or below .85?</td>
</tr>
<tr>
<td>16</td>
<td>Student 1</td>
<td>Above</td>
</tr>
</tbody>
</table>

An example of interactions between two thermodynamics students and the agent is provided in Table 2. In recent work, we have successfully used the same technology to support interactions between larger student groups, such as 3-5 person teams. As in the example of Table 1, in Table 2, Tutor is the chat agent. As we see from Table 2, there are two main interaction components working behind the tutor agent: the Hinting Actor and the Tutoring Actor. These components are supported in their task by two filtering components: the Hinting Filter and the Tutoring Filter. The role of the Hinting Filter is to look for a student contribution related to an underlying concept. When such a contribution (Line 1) is detected, the Hinting Actor generates a hint pointing the students to a relevant page in the instruction book (Line 2). Note that what is considered relevant is based on the agent’s analysis of the ongoing discussion, detecting which topics have been raised by the students so that support related to those topics can be triggered. In the example in Table 2 we see that Student 1 has raised the issue of increasing pressure to the connector, which is what is known as the maximum pressure of the cycle. This is an important parameter in the design of the power plant, and there are principles related to decisions about its...
configuration that students should keep in mind. Also, the Tutoring Actor invites the students to ask for help on the concept (Line 3). This is how the agent offers the student the opportunity to exercise some control over the timing of the help that is offered by the conversational agents. The Tutoring Filter detects the request for help (Line 4) and initiates an instructional dialog with the students (Lines 5 through 16). Note that the dialogue that is initiated is pertinent to the conversation about manipulating the maximum pressure of the cycle.

Integrated Mathematics and Engineering Education
Software agent-monitored tutorials are also being created to reinforce mathematics fundamentals as part of the undergraduate engineering curriculum. It is widely accepted that mathematics would be better taught to engineers in a contextualized manner where they would have the opportunity to learn it in the context of solving engineering problems. However, engineering and science curricula are typically structured in such a way that the mathematics curriculum is offered in a largely decontextualized way so that it can be offered in a uniform, cost effective manner, to students in a variety of disciplines.

Since the mid-80s, research in increasing success with university level mathematics and retention in engineering majors has drawn from the literature on collaborative learning, especially in connection with motivation and self-efficacy issues. Collaborative learning, classroom discussion, and development of math explanation skills have also traditionally been a focus within the math reform movements in the field of education. Research in problem-based learning at the college level supports the importance of a facilitator’s role in keeping the discussion moving in a productive direction. In our prior work developing and evaluating an environment to support collaborative calculus problem solving, computer based prompts played the role of an instructor, improving the interaction, and leading to significant learning gains in comparison to a non-supported control condition.

Beginning in 2003, Wright State University undertook a systematic evaluation and reformation of how they treat engineering mathematics. Their efforts were primarily motivated by retention problems within their engineering program. Historically, only about 40% of their students initially interested in an engineering or computer science degree would advance past the freshman-level calculus sequence. Student surveys suggested that the traditional mathematics curriculum was perceived as very difficult, and more importantly, not tied to applications. Students perceived the math sequence as representative of the overall content of their chosen majors, and would choose another career path, leading many well-qualified students away from STEM-based careers.

The solution to this problem developed at Wright State contains 3 principal characteristics:

1) Development of a first-year engineering math course, EGR 101, taught by engineering faculty, that covers and motivates only math topics actually used in later engineering courses.

2) Restructuring of the engineering curriculum so that EGR 101 is the only math prerequisite for subsequent engineering courses.

3) Retaining a math sequence in concert with ABET accreditation requirements, still taught by the math department, but shifted to later years in the undergraduate curriculum.
The Wright State Model for engineering mathematics education has many advantages. It tackles the retention problem directly, by introducing engineering mathematics in a motivational, physically-based first-year course. Because students see mathematics in the context of applications from the start, the EGR 101 course also sets the stage for better retention of topics covered in subsequent math fundamentals courses. Logistically, the approach is appealing because it does not replace the traditional math sequence taught by the math department. It simply removes its role as a prerequisite for subsequent engineering courses, allowing math courses to be taken later in the engineering curriculum. Because of these features, the Wright State Model has been highly successful not only at Wright State, but at other universities as well.

Despite the success of the WSU Model, and its straightforward approach to the long-standing problem of a lack of integration between math fundamentals and engineering applications, it can’t be expected to work within all engineering programs. Many engineering programs may not be able to offer an additional mathematics course in the freshman year. Depending on the program, other material may need to be covered in that time frame. Second, many engineering programs do not have the retention-related concerns tied to the math sequence that Wright State has had. The pedagogical value of linking math with engineering applications may be the principal concern. It is also worth noting that the WSU Model, though a systemic reform, takes a traditional approach of offering additional course and lab content. It does not attempt to exploit rapid advances in educational technology to achieve its goals. An approach capturing many of the features of the WSU Model without the need for an additional first-year mathematics course would be a major advantage. The goal of current work by the authors is to do this, exploiting their agent-guided learning technology to create an alternative delivery mechanism for many features of the WSU Model.

The principal outcome of this work will be a series of dialogue-based web course modules appropriate for use in a first- and second-year mechanical engineering courses and pre-college outreach programs. Consistent with their timing, projects and course modules will reinforce math fundamentals taught in the freshman/sophomore math sequence, exploiting the self-discovery and dynamic interaction features of the tutorial interface. We will experiment with a wide range of methods for instructor and teaching assistant monitoring of student work. This can be done remotely, by monitoring student dialogues digitally, or directly by holding more traditional help sessions run by course instructors or assistants. We will explore methods for optimizing the levels of direct personal, remote personal and purely agent-based interactions most effective for our students. In the extreme, students will use tutorials in the evenings or at other times convenient to them, working with fellow students with only the dialogue agent monitoring and coaching their work. If this proves practical, students could obtain the benefits of an engaging, collaborative mathematical modeling experience fully on their own.

**Results to Date**

We have run a variety of pilot studies with the systems described above in undergraduate engineering courses to verify the effectiveness of the instructional support we offer. For example, in one study, where students worked in pairs on a power plant design task, students who worked in pairs that included the support of a conversational agent learned 1.24 standard deviations more than control condition students who worked alone in the same learning environment but without a partner or the conversational agent. Students who worked with a
partner student achieved a learning outcome of 0.9 standard deviations compared to the control condition and students working with only a conversational agent learned 1.06 standard deviations more than the control case. Note that one standard deviation is equivalent to a full letter grade.

In subsequent studies\textsuperscript{10, 16, 17} we continually refined our conversational agent technology, achieving even better learning outcomes compared to our previous agents. Using various interaction tactics\textsuperscript{27} such as Attention-Grabbing\textsuperscript{16} and Ask-When-Ready we were able to improve learning outcomes by ensuring that the students are paying attention to the conversational agent before the agent starts discussing an engineering concept with the students. An example of the Ask-When-Ready tactic is shown in Table 1.

While these earlier experiments focused on achieving attention from the student during the course of the interaction, recent experiments have focused on the use of interaction strategies that could manage student attention more holistically during the entire interaction. Inspired by research in small group communication which finds that group interaction includes both task-related (instructional) as well as social processes, we have developed agents that can display social behavior at appropriate instances during the interaction. Using 11 social interaction strategies\textsuperscript{10}, we were able to achieve an additional 0.71 standard deviations improvement over our best agents that perform only instructional behavior.

Besides the learning outcomes, we observed improvements in several perception measures such as likeability, friendliness and group bonding, as shown in Figure 6. In the figure, which presents results of student surveys, ranking various aspects of their experience on a scale of 1 to 7 (see Table 3), the Task condition corresponds to the agents which did not display any social behavior. The Social condition corresponds to agents with social interaction capabilities and the Human condition corresponds to the use of human tutors to display the social behaviors. The improvements of the Social condition over the Task condition are not as pronounced as those for the Human condition. This suggests scope for further improvement of our agents by making them more human-like in their social capabilities.

\textbf{Figure 6}: Middle school students learning about CNC machining as part of the outreach project
Table 3: Items on the Perception Survey

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Likeable</td>
<td>I liked the tutor very much.</td>
</tr>
<tr>
<td>Friendly</td>
<td>The tutor was very cordial and friendly during the discussion</td>
</tr>
<tr>
<td>Ideas</td>
<td>The tutor was providing very good ideas for the discussion</td>
</tr>
<tr>
<td>Tension-Release</td>
<td>The tutor kept the discussion at a very comfortable level socially</td>
</tr>
<tr>
<td>Group Bonding</td>
<td>The tutor was part of my team</td>
</tr>
<tr>
<td>Agreed</td>
<td>The tutor received the ideas and suggestions I contributed to the discussion positively</td>
</tr>
<tr>
<td>Discussion Satisfaction</td>
<td>I am happy with the discussion we had during the design challenge</td>
</tr>
<tr>
<td>Task Satisfaction</td>
<td>My group was successful at meeting the goals of the design challenge</td>
</tr>
<tr>
<td>Legitimacy</td>
<td>The design challenge was exciting and I did my best to come up with good designs</td>
</tr>
</tbody>
</table>

Following these observations, our recent investigations have focused on determining the right amount of social behavior and improving the timing of these behaviors displayed by the agents. We have found that the displaying the right amount of social behavior\(^\text{17}\) is critical to achieving improvements in outcomes. Excessive display of social behavior becomes a distraction and fails to achieve an improvement over a condition where no social behavior is performed.

**Conclusions**

Agent-monitored collaborative tutorials are being integrated into software-based mechanical engineering projects at Carnegie Mellon. Thus far, projects at the freshman and sophomore levels have been deployed within the undergraduate curriculum, with a spin-off of the freshman project also used as part of a middle school outreach workshop. Work is also underway to use this approach to enhance mathematics instruction within the curriculum. This platform is being used to explore the effectiveness of a variety of collaborative environments, as well as combinations of collaborative learning, automated instruction, classroom-style instruction and interpersonal instruction. Tutor agents are also being used to quantify the importance of social interaction (both simulated via tutors and actual via teaching assistants) in learning. The goal is to not only more effectively teach engineering fundamentals in the context of software use, but to also increase the efficiency of integrating software projects into courses.

**Acknowledgments**

This research was supported by the National Science Foundation Division of Undergraduate Education (DUE) Course Curriculum and Laboratory Improvement (CCLI) Program and the Engineering Education and Centers (EEC) Innovations in Engineering Education, Curriculum and Infrastructure (IEECI) Program under grants DUE-0837661, DUE-1022958 and EEC-0935145. This research is also supported through a software grant from Parametric Technology Corporation (PTC). The authors would also like to thank their collaborators, Ann Shoplik and Pamela Piskurich of the Carnegie Mellon C-MITES program, Jack Zhou of Drexel University, Nathan Klingbeil of Wright State University, Glenn Beltz of the University of California, Santa Barbara, and John Stuart of PTC.
References


21. Treisman, P. M. (1985). A Study of the Mathematics Performance of Black Students at the University of California, Berkeley. PhD Dissertation from the Graduate Division of the University of California at Berkeley, program of Science/Mathematics Education.


