Walker, E., McLaren, B. M., Rummel, N., & Koedinger, K. R. (2007). Who Says Three's a Crowd? Using a Cognitive Tutor to Support Peer Tutoring. In the *Proceedings of the 13th International Conference on Artificial Intelligence in Education* (AIED 2007), 399-406.

Who Says Three's a Crowd? Using a Cognitive Tutor to Support Peer Tutoring

Erin WALKER[1], Bruce M. MCLAREN[1], Nikol RUMMEL[2], Ken KOEDINGER[1]
[1]*Human Computer Interaction Institute, Carnegie Mellon University, Pittsburgh, USA*
[2]*Institute of Psychology, University of Freiburg, Freiburg, Germany*
{erinwalk, koedinger, bmclaren}@cs.cmu.edu; rummel@psychologie.uni-freiburg.de}

Abstract. Adding student collaboration to an intelligent tutoring system could leverage the benefits of both approaches. We have incorporated a mutual peer tutoring script, where students of similar abilities take turns tutoring each other, into the Cognitive Tutor Algebra. In this paper, we identify three design principles for peer tutoring, and discuss how they were realized in our peer tutoring script. We then develop a cognitive model for peer tutoring, and drawing from student data, identify places for an intelligent tutor to provide feedback. Finally, we describe the implementation of the script and our plans for formal evaluation.

Keywords. Peer tutoring, intelligent tutoring systems, collaborative learning

Introduction

Augmenting an intelligent tutoring system (ITS) with collaborative learning activities holds the promise of increasing the benefits of the ITS. The structured problem-solving and individualized feedback provided by an ITS such as the Cognitive Tutor Algebra (CTA) has increased student learning by approximately one standard deviation over traditional classroom instruction [1]. Critics of this approach argue that students acquire shallow knowledge while using the ITS, since there is not much freedom for students to construct their own knowledge or learning path. On the other hand, collaboration has also been shown to have positive effects on learning, and group activities encourage students to develop deep knowledge [2]. However, these activities are only effective at increasing learning if students interact in positive ways [3]. Students typically do not collaborate well spontaneously and often do not receive enough guidance from their teacher. The CTA curriculum includes collaborative work, but these activities are often not optimally administered by teachers or followed by students [4]. To facilitate appropriate interaction, researchers develop collaboration scripts in which participating students are given specific roles and activities [5]. An ITS can provide a platform for implementing a script and adaptive support for students following the script.

If ITS methods can be applied to tutoring collaborative activities, the benefits of both approaches could be combined and strengthened. Some preliminary work has been done in this area: Harrer et al. [6] have integrated the collaborative tool Cool Modes with the Cognitive Tutor Authoring Tools (CTAT), an authoring tool for tutoring systems, and Soller [7] has created a model of structured chat in order to develop an ITS to improve student interaction. Another approach is to use an intelligent agent as one of the collaborators. Biswas et al. [8] had students, helped by a mentoring agent, teach an intelligent agent about ecosystems. Our approach involves incorporating peer tutoring into the CTA. We designed a peer tutoring addition that draws from previous successful peer tutoring scripts, and based on a task analysis and empirical data, developed a preliminary model for peer tutoring and adaptive cognitive support. We have implemented the extension and will soon be conducting a formal evaluation.

1. Peer Tutoring Addition to the Cognitive Tutor Algebra

In our peer tutoring addition to the CTA, we adopted a reciprocal peer tutoring script, where students of similar abilities take turns tutoring each other on course material. This type of peer tutoring has been shown to increase mathematics learning in a realistic classroom environment [9]. In order to explore the factors that make peer tutoring an effective learning intervention for both the tutor and the tutee, researchers have scripted the peer tutoring process to encourage students to behave in particular ways, and then compared the scripted condition to an unscripted control. In conditions where tutors are encouraged to provide elaborated explanations [10], set goals for tutoring and monitor the skills being acquired [11], and prepare ahead of time [9], peer tutors tend to learn more. Researchers have also explored the effect of the peer tutor on the tutee by coding tutoring transcripts for particular behaviors, and correlating those with learning. Tutees learn more when they request help, tutors respond with deep explanations, and tutees apply the explanations to their problem-solving [12].

Biswas et al. [8] identified three aspects of interaction that exist in learning by teaching: taking responsibility for actions, reflecting on knowledge, and developing structured knowledge. These processes are also present in successful peer tutoring. When students prepare, they take responsibility for knowledge because they will soon be communicating it to another. Students also must monitor their tutee's knowledge, becoming more aware of problem skills and identifying gaps in their own knowledge. Lastly, during tutoring, students must ask questions and receive explanations, leading them to better structure their knowledge. From this empirical and theoretical literature, we have derived three design principles for our peer tutoring script. Before tutoring, students should complete exercises on both the skills required to solve domain problems and the skills required to teach them (*preparation*). As a tutor, students should set goals for their partner and monitor their partner's progress (*reflection*). Students should talk about the domain in elaborated and specific ways (*interaction*).

Our peer tutoring script is built on an equation solving unit of the CTA, where students are given a prompt (e.g., "Solve for x") and an equation (e.g., "ax + bx =c"). To implement the *Preparation* principle, we divided student activity into two phases: a preparation phase, where students solve problems using the CTA, and a collaboration phase, where students tutor each other on the problems they solved. In the preparation phase, students solve problems using an equation solver tool and receive feedback from the cognitive tutor. Their skill mastery is displayed in a "skillometer". After each problem, they read descriptions of good tutoring behavior and then answer questions such as “What would be a good hint to give to your partner?”

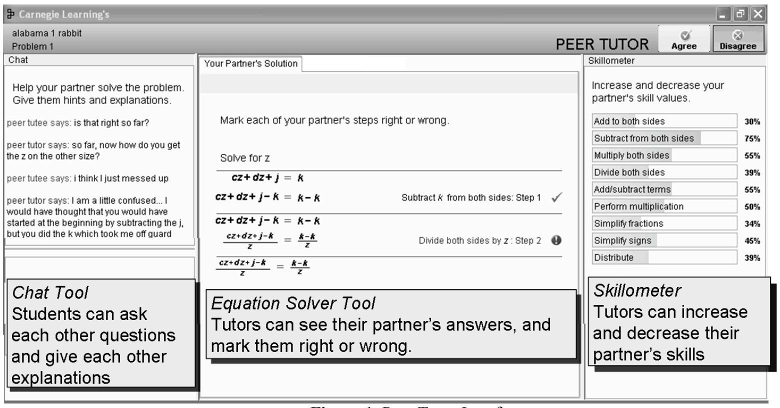

Figure 1. Peer Tutor Interface

During the collaboration phase, students are put in same-ability groups and collaborate at different computers in the same classroom. They take turns being the tutor and tutee. See Figure 1 for a screenshot of the peer tutor’s interface. The tutee still solves the problem with the equation solver, as if using the regular ITS. Peer tutors can see the tutee’s actions, but cannot solve the problem themselves. Instead, to incorporate the *Reflection* principle, the peer tutor can mark the tutee’s answers in the equation solver and adjust the tutee’s skills in the skillometer. These actions, seen by the tutee, prompt students to reflect on skills required to solve the problem and how close the tutee is to mastering those skills. Students also discuss the problem in a chat window, following the *Interaction* principle.

As an illustration of our approach, the following is an excerpt from a real tutor-tutee interaction. The students were solving the problem “$cz + dz + j = k$” for z. The tutee subtracts k from both sides in the equation solver, and then asks in the chat, “Is that right so far?” The tutor can see the tutee’s action, and responds incorrectly: “So far, now how do you get the z on the other side?” The tutee divides by z, and realizes something is wrong, typing “I think I just messed up.” The tutor responds, “I am a little confused… I would have thought that you would have started at the beginning by subtracting the j, but u did the k which took me off guard.” The tutor then marks the step wrong in the equation solver. The tutee, seeing the tutor’s feedback, undoes the incorrect steps and takes the correct step, subtracting both sides by j. The tutor is satisfied, and increases the value of the “Subtract both sides” skill bar. Once the students have solved the problem, the tutee clicks the done button, the tutor agrees that they are done, and the students move to the next problem and switch roles.

This interaction comes from a pilot study in which we explored how students used the script without any adaptive support (see [13]). We can use this data to determine where adaptive support might be most effective during peer tutoring. The script was implemented as an addition to the CTA, and piloted using 20 students from two Algebra-1 classes. We found that students did learn from the peer tutoring, and appeared to be engaged in preparing, interacting, and reflecting on their knowledge. However, peer tutors struggled to provide tutees with answers, despite having already solved the problems and having been given printouts of the answers. They did not complete many problems, skipped problems without completing them, and relied too heavily on teacher assistance. These behaviors are undesirable because no matter how well students are collaborating, fewer problems successfully completed means fewer opportunities to master domain skills. Targeting intelligent tutoring support towards the cognitive aspects of peer tutoring may provide students with greater benefit from positive interaction, as well as leading to more problems successfully completed.

2. Cognitive Model for Peer Tutoring

Because of the results of the pilot study, we decided to focus our attention on providing adaptive support for peer tutors in knowing how to solve the problem they are tutoring and being able to communicate that knowledge to their partner. Further supporting the design principles that we identified, as do many other models of peer tutoring incorporated in peer tutoring scripts, would not be helpful without first ensuring that the peer tutor is capable of correctly advising the tutee.

2.1. Rational Task Analysis

To evaluate whether peer tutors are doing their job well and to support them in performing the tutoring task, we have designed a model of peer tutoring (Figure 2). For simplicity, our model is constrained in three ways. First, it assumes that peer tutor and tutee actions are synchronous, in that every action by the tutee is followed immediately by a tutor response. In practice, this is unlikely and may be undesirable; a fast tutee should not be held up by a slow tutor. Second, our model assumes that certain actions will always be taken even if those actions have been made redundant by other actions. For example, we have the peer tutor mark an answer incorrect before giving the peer tutee feedback. In a real situation, the peer tutor might simply tell the tutee that their answer is incorrect while giving them feedback, and therefore would not need to explicitly mark it. Our model treats this single action as two separate actions. Finally, our model focuses on the steps of the tutorial process, rather than the content. We are not modeling what should occur during discussion, but *when* discussion should occur. Once we can successfully support the tutoring process, we will focus on the content.

The model starts when the students are in a state of “working on the problem”. The peer tutee can take one of three actions: take a problem step, ask for help from the peer tutor, or indicate they are done. The peer tutor can also start the model by determining that the peer tutee needs help. The model can then be divided into three general types of peer tutor responses: correction (bold boxes), skill assessment (dashed boxes), and discussion (regular boxes). The model assumes that correction is the immediate response to the tutee taking a step or selecting done. If the tutee action is correct, the tutor should mark it right. If the step or action is incorrect, the tutor should mark it wrong. The peer tutor starts a discussion whenever the tutee needs help or feedback, which may be the case after an incorrect answer by the tutee, when the tutee requests help, or when the tutee appears to be struggling. We have divided the discussion into three substeps: Initiation (“Tutor starts discussion”), the bulk of the discussion (“Students discuss step”), and termination (“Tutee understands?”). The peer tutor engages in skill assessment after correction and discussion actions, adjusting the tutee's skill bars as appropriate. Skill assessment is the lowest priority; the tutee needs feedback in order to continue but can move to the next step as the tutor adjusts skills.

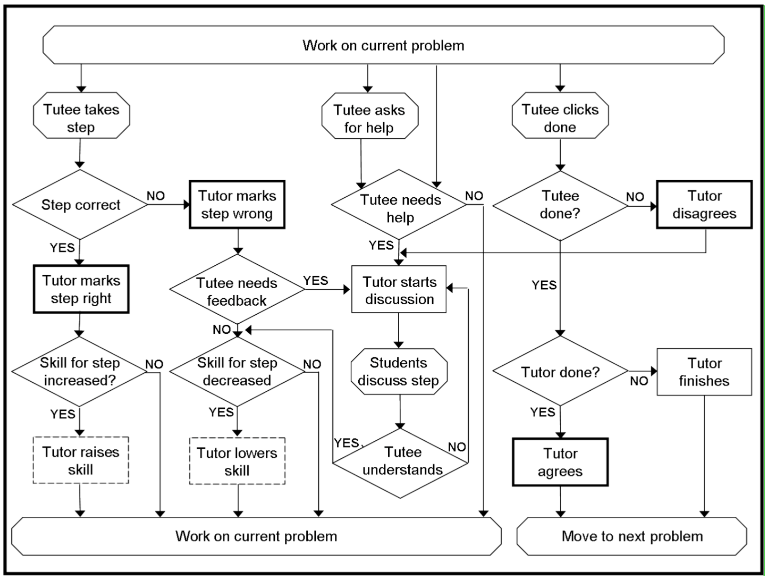

Figure 2. "Ideal" model of peer tutoring. Bold boxes are correction actions, dashed boxes are skill assessment actions, and regular boxes are discussion actions.

Office 2004 Test Drive…, 3/7/07 11:09 AM
Comment: I shrunk the size of the figure. I don't think it needs to be so big, and it went outside of the margin. Also, perhaps this will help to cut the paper back to 8 pages.

2.2. Design of Hints and Feedback

Drawing from the data collected in our pilot [13], we designed preliminary ITS support for peer tutoring based on two principles. We noticed a lot of interaction in the chat window between the peer tutor and the tutee; when stuck, tutees would use the chat to ask tutors for help, and tutors would use the chat not only for explanations, but to provide tutees with feedback on whether steps were right or wrong. To avoid disrupting this interaction, communication between the peer tutee and the computer tutor is mediated by the peer tutor. To get a hint, the tutee requests one from the peer tutor, and the peer tutor then requests one from the computer tutor. Any feedback messages given by the computer tutor are displayed to the peer tutor, who then explains the message to the tutee. For similar reasons, the computer tutor provides feedback to the peer tutor based on action, not inaction. If the peer tutor marks a wrong answer right, the computer tutor will step in, but if the peer tutor fails to correct an action, the computer tutor will not intervene, as it is possible that students are having a productive discussion or the peer tutor is occupied with another problem step. These hints and feedback are designed to help students complete problems correctly, so that they will benefit more from positive interaction (following the *Interaction* principle).

In our pilot, students in one class completed on average less than five problems during the collaborative session, while students in the other class only completed roughly 60% of the problems they attempted. Providing feedback on peer tutor correction actions might help students to correctly complete more problems. When peer tutors are wrong in a correction action, it is highlighted in the interface, giving them a visual indication that they have made a mistake. If peer tutors mark a correct problem step incorrect, they get a message like, "Your partner has the right idea. Ask them why they took that step." If peer tutors mark an incorrect problem step correct, he or she gets a more complicated message involving the computer tutor feedback the tutee would have received (e.g., "It is better to divide by -7.1, since that would leave *y* and not –*y*"), and a prompt to collaborate like "Explain to your partner what to do and why." Similarly, if peer tutors agree with an incorrect tutee done action, they are informed that the action is incorrect, and students are not moved to the next problem.

During the pilot, we noticed that peer tutees would ask tutors for hints, and tutors would be unable to provide guidance, saying "I don't know" in response to questions or waiting for teacher help. To give the peer tutor extra assistance, we now allow them to ask the computer tutor for a hint. The hint, which has a cognitive and collaborative component, is then delivered only to the peer tutor. The cognitive component involves the advice the computer tutor would have given to the tutee during a typical tutoring session at the problem step the tutee is currently on, such as "-7.1*y* is -7.1 times *y*. How do you undo multiplication?" If the computer tutor advice is multi-level, the current hint is multi-level. The collaborative component involves a prompt to discuss feedback with their partner, such as "Talk to your partner about how this hint applies to the problem."

Finally, during the pilot, we noticed that peer tutors would tend to raise their partner's skill bars all the way to the maximum during the first or second problem. In an attempt to encourage better reflection on problem skills (following the *Reflection* principle), we send the tutor a feedback message if they try to raise a skill bar more than 15% per problem that reads, "Slow down! Before increasing more, wait until your partner has shown this skill on another problem." Students receive a similar message if they try to lower a skill bar more than 15%. In the future, we hope to have skill bar feedback that tutors more specifically which skill the peer tutor manipulates. We also intend to tutor the chat discussion. However, we believe the correction tutoring we have designed will be effective in itself at supporting peer tutors as they instruct their tutees.

3. Implementation of Peer Tutoring Script with Cognitive Tutoring Support

To implement the peer tutoring script, we modified the standard CTA architecture. The CTA separates the interface to the user (*tool*) from the intelligence (*tutor*) and from a central student model (*learner management*). We changed this architecture to enable multiple tool and tutor components. With this multi-tool capacity, the peer tutor and tutee can interact using separate interfaces at separate computers. With the capacity for multiple tutors, other tutoring modules such as a collaborative tutor can be added (see Figure 3a). To enable multiple components in different configurations, we changed the standard CTA components to function independently and remotely (see [14]). We then expanded the control module of the CTA to include a *mediator,* which intercepts all messages sent by a component and directs them to appropriate targets. The mediator maintains a list of session types, containing tools, tutors, and instructions for how messages are passed. The most basic session type contains the standard CTA tool and the cognitive tutor. The peer tutoring session type has a peer tutee tool, a peer tutor tool, and an echoing agent which facilitates collaboration by echoing a user action on one tool to the other tool's interface.

The first step to incorporating computer tutor feedback into our addition to the CTA was to develop a separate tutor module, called the correction tutor. The correction tutor has knowledge of peer tutoring actions instead of domain knowledge. The expert model of the correction tutor is based on a simple bug rule: If the peer tutor response to a tutee step does not match the cognitive tutor response, the bug rule fires, and the correction tutor highlights the problem step and displays feedback on the peer tutor's screen. The peer tutor must have responded to the step before the rule can fire, and thus the model does not force the peer tutor to respond to every step. The model also has a bu [illegible] mar

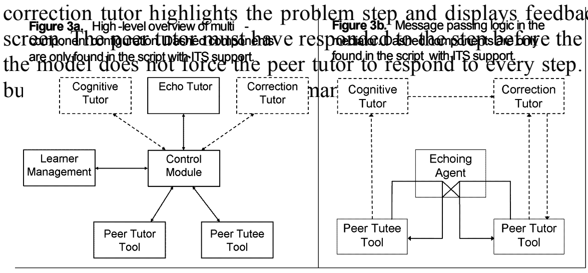

Figure 3. Changes to Cognitive Tutor Algebra architecture for peer tutoring script.

After implementing this component, we defined a new session type for "peer plus cognitive tutoring" by modifying the existing peer tutoring session type. We added two tutors to the list (the "Cognitive Tutor" and the "Correction Tutor"). Figure 3b diagrams the message passing logic within the mediator for peer plus cognitive tutoring. In this configuration, when the peer tutee takes an action, the echoing agent sends the action to the peer tutor's screen. In addition, the cognitive tutor evaluates the action, and sends the evaluation to the correction tutor. When the peer tutor takes an action, it is sent to the echo tutor, which echoes the action onto the peer tutee's screen, and to the correction tutor, which compares the peer tutor evaluation to the cognitive tutor evaluation. If responses do not match up, the correction tutor sends feedback to the peer tutor. The peer tutor can also request a hint from the correction tutor, which has stored the cognitive tutor hint for that step, and delivers it to the peer tutor.

4. Conclusions and Plans for Evaluation

We have designed and implemented a peer tutoring addition to an ITS, where students take turns tutoring each other and the ITS provides tutoring support. Our next step is to conduct a second experiment to evaluate the enhanced peer tutoring script. We will compare three conditions: the enhanced peer tutoring script (peer plus cognitive tutoring condition), the baseline peer tutoring script (peer tutoring condition), and a control where students use the ITS individually (*i*ndividual condition). We expect that students in the collaborative conditions will learn more than students in the individual conditions and students in the peer plus cognitive condition will learn the most. Learning will be measured by items from the unit, as well as transfer items that evaluate students' abilities to apply their knowledge and explain their solutions.

This work is an initial step toward combining the benefits of intelligent tutoring systems and collaborative learning activities. We have modified the Cognitive Tutor Algebra so that students can use it collaboratively to tutor each other while being provided with cognitive support by the system. Our ultimate goal is to extend our implementation to include collaborative tutoring of student interaction in order to increase deep learning.

References

[1] Koedinger, K. R., Anderson, J. R., Hadley, W. H., & Mark, M. A. (1997). Intelligent tutoring goes to school in the big city. International Journal of Artificial Intelligence in Education, 8, 30-43.

[2] Gokhale, A. (1995). Collaborative learning enhances critical thinking. Journal of Technology Education, 7(1), 22-30.

[3] Johnson, D. W. and Johnson, R. T. (1990). Cooperative learning and achievement. In S. Sharan (Ed.), Cooperative learning: Theory and research (pp. 23-37). New York: Praeger.

[4] Ritter, S., Blessing, S. B., & Hadley, W. S. (2002). SBIR Phase I Final Report 2002. Department of Education. Department of Education RFP ED: 84-305S.

[5] Kollar, I., Fischer, F., & Hesse, F. W. (in press). Collaboration scripts - a conceptual analysis. Educational Psychology Review.

[6] Harrer, A., McLaren, B. M., Walker, E., Bollen, L., and Sewall, J. (2006). Creating Cognitive Tutors for Collaborative Learning: Steps Toward Realization. User Modeling and User-Adapted Interaction: The Journal of Personalization Research (UMUAI), vol 16, nos.3-4, 175-209.

[7] Soller, A.L. (2001). Supporting Social Interaction in an Intelligent Collaborative Learning System. International Journal of Artificial Intelligence in Education, 12, 40-62.

[8] Biswas, G., Schwartz, D. L., Leelawong, K., Vye, N., & TAG-V. (2005). Learning by teaching: A new agent paradigm for educational software. Applied Artificial Intelligence, 19, 363–392.

[9] Fantuzzo, J. W., Riggio, R. E., Connelly, S., & Dimeff, L. A. (1992). Effects of reciprocal peer tutoring on mathematics and school adjustment: A componential analysis. Journal of Educational Psychology, 84, 331-339.

[10] King, A., Staffieri, A., & Adelgais, A. (1998). Mutual peer tutoring: Effects of structuring tutorial interaction to scaffold peer learning. *Journal of Educational Psychology, 90*, 134-152.

[11] Fuchs, L.S., Fuchs, D., Prentice, K., Burch, M., Hamlett, C.L., Owen, R., & Schroeter, K. (2003). Enhancing third-grade students' mathematical problem solving with self-regulated learning strategies.

[12] Webb, N.M., Troper, J.D., & Fall, R. (1995). Constructive activity and learning in collaborative small groups. Journal of Educational Psychology, 87, 406-423.

[13] Walker, E., Rummel, N., McLaren, B. M., and Koedinger. K. R. The Student Becomes the Master: Peer Tutoring in the Cognitive Tutor Algebra. To appear as a short paper to CSCL07.

[14] Walker, E., Koedinger, K. R., McLaren, B. M. and Rummel, N. (2006). Cognitive Tutors as Research Platforms: Extending an Established Tutoring System for Collaborative and Metacognitive Experimentation. Proceedings of the 8th International Conference on Intelligent Tutoring Systems.

Acknowledgment

This research is supported by the Pittsburgh Science of Learning Center, NSF Grant # 0354420.