Research and Practice in Technology Enhanced Learning

INTEGRATING COLLABORATION AND INTELLIGENT TUTORING DATA IN EVALUATION OF A RECIPROCAL PEER TUTORING ENVIRONMENT

ERIN WALKER

Human-Computer Interaction Institute, Carnegie Mellon University, 5000 Forbes Ave.
Pittsburgh, Pennsylvania, 15213, USA
erinwalk@andrew.cmu.edu

NIKOL RUMMEL

Institute of Psychology, University of Freiburg, Engelbergerst. 41,
79085 Freiburg, Germany
rummel@psychologie.uni-freiburg.de

KENNETH R. KOEDINGER

Human-Computer Interaction Institute, Carnegie Mellon University, 5000 Forbes Ave.
Pittsburgh, Pennsylvania, 15213, USA
koedinger@cmu.edu

Intelligent tutoring systems have been successful at increasing student mathematics learning, but may be further improved with the addition of collaborative activities. We have extended the Cognitive Tutor Algebra, a successful intelligent tutoring system for individual learning, with a reciprocal peer tutoring activity designed to increase conceptual learning. While using our peer tutoring environment, students take on tutor and tutee roles, and engage in both problem-solving actions and dialogue. In a classroom study, we randomly assigned 62 participants to three conditions (adaptive assistance to peer tutoring, fixed assistance to peer tutoring, and individual learning). All conditions yielded significant learning gains, but there were no differences between conditions in final outcomes. There were significant process differences, however. We assessed student interaction using problem-solving information logged by the intelligent tutoring system and collaborative dialogue captured in a chat window. Our analysis integrated these multiple data sources in order to better understand how collaborative dialogue and problem-solving actions might lead to conceptual learning. This rich data sheds light on how students benefitted from the reciprocal peer tutoring activity: Peer tutors learned when they reflected on tutee problem-solving actions, and tutees learned when the tutor help was responsive to those actions.

Keywords: adaptive collaborative learning support, peer tutoring, in vivo experiment

1. Introduction

In order to improve mathematical problem-solving skills, it is important to increase student conceptual understanding, which is crucial for the transfer of learned skills to new problems (e.g., Rittle-Johnson & Alibali, 1999). In our work, we combine two different interventions in an attempt to improve student conceptual math learning: intelligent tutoring systems and computer-supported collaboration. Intelligent tutoring systems are systems that compare student actions to a domain model in order to provide context-sensitive help on problem-solving steps or solutions and adapt instructional activities to student needs (VanLehn, 2006). Intelligent tutors, like the Cognitive Tutors developed at Carnegie Mellon University, have been successful at increasing math learning in the classroom by approximately one standard deviation over traditional instruction (Koedinger, Anderson, Hadley, & Mark, 1997). However, the impact of these systems still falls short of the effects achieved by expert human tutors (Bloom, 1984). Augmenting intelligent tutoring systems with activities that encourage monitoring and elaboration such as self-explanation (e.g., Aleven & Koedinger, 2002, Weerasinghe & Mitrovic, 2006), reflection (e.g., Mitrovic & Martin, 2002) and help-seeking (e.g., Roll, Aleven, McLaren, & Koedinger, 2007) may further improve the deep conceptual learning of students. In particular, we are interested in augmenting intelligent tutoring systems with computer-supported collaborative activities. Collaboration has been demonstrated to have a positive effect on individual and group learning outcomes (Lou, Abrami, d'Apollonia, 2001), and in particular to promote deep elaboration of the learning content (Teasley & Fischer, 2008). Our work investigates how collaborative learning with intelligent tutoring support might be more effective at improving domain learning than individual learning with intelligent tutoring support.

Simply having students collaborate while using an intelligent tutoring system is unlikely to be effective. The positive effects of collaboration do not emerge spontaneously, but require the careful structuring of the collaboration so that particular promotive interactions emerge (Johnson & Johnson, 1990). One effective way of doing so is by scripting the collaboration, that is, by designating roles and activities for students to follow (Kollar, Fischer, & Hesse, 2006). Traditionally, computer-supported collaboration scripts are *fixed* in that they provide the same level of support for all students. This level of support may be unnecessary for students who have good internal collaboration scripts or who are already experienced collaborators (Kollar, Fischer, & Slotta, 2005), and thus may overstructure collaboration for many students, decreasing student control over the learning environment and motivation (Dillenbourg, 2002). On the other hand, the same scripts may provide insufficient support for poor collaborators, who, without adequate monitoring and feedback, often do not execute collaborative activities as planned (Ritter, Blessing, & Hadley, 2002). Using intelligent tutoring technology to assess student collaboration as it occurs and provide assistance when needed may therefore improve on fixed techniques. Early results in adaptive collaborative learning support are promising, demonstrating learning improvements over unsupported collaboration (Gweon, Rosé, Carey, & Zaiss, 2006) and interaction benefits of adding collaboration to intelligent tutoring (Diziol et al., 2008). However, work on these systems is generally still at an early stage (see Soller, Jermann, Mühlenbrock, & Martinez, 2005, for a review), and few existing systems have been evaluated for their impact on learning. Thus, a second research question we investigate in this paper is whether providing adaptive support to collaboration is indeed more effective than providing fixed support.

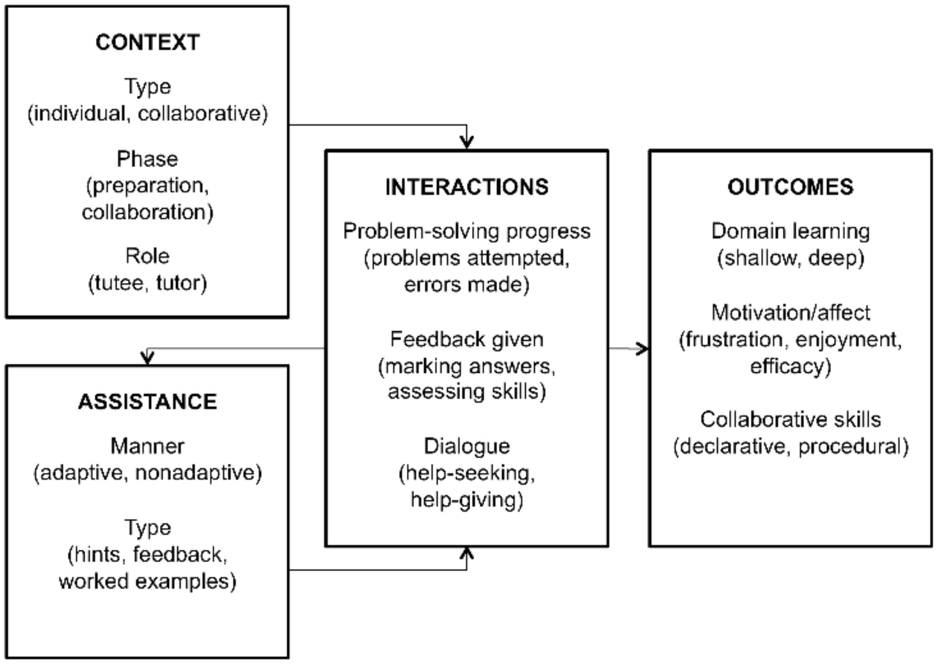

Figure 1. Types of data collected in our study. Both the CONTEXT of the learning environment and the ASSISTANCE it provides influence student INTERACTIONS. Particular student INTERACTIONS may yield changes in the manner, type or content of the ASSISTANCE provided by the system. The nature and number of student INTERACTIONS then influence learning and motivation OUTCOMES.

To address our research questions regarding the integration of intelligent tutoring systems and computer-supported collaboration scripts, we have augmented an existing intelligent tutoring system, the Cognitive Tutor Algebra (CTA), with a reciprocal peer tutoring activity. The resulting system logs fine-grained problem-solving data like an intelligent tutoring system, but also automatically records collaborative dialogs. In this paper, we demonstrate that by integrating and analyzing both types of data we can better explore the effects of adaptive collaborative learning support on learning. In the background section, we describe the individual version of the CTA, present the reciprocal peer tutoring background for our intervention, describe related work on adaptive support for peer tutoring, and then discuss in more detail the types of data our system produces. Next, in the methods section of the paper, we describe a study where we compared the effects of peer tutoring with adaptive assistance to peer tutoring with fixed assistance and individual use of the CTA. Students in our study engage in different types of activities, go through different phases, and take on different roles (see the “Context” box of Figure 1). Additionally, they receive either adaptive or fixed assistance (see the “Assistance box of Figure 1). Our dependent measures spanned learning outcome data, interaction data, and problem-solving data (see the “Outcomes” and “Interactions” boxes of Figure 1). In our study, the picture of the results formed by combining the different measures gave us a greater understanding of what was occurring than each individual measure. We discuss the implications of our results for technology enhanced learning research and practice. We also comment on relevant issues of data analysis methods and on using multiple measures to build a rich picture of factors that may lead to robust student learning.

2. Background

2.1. *The Cognitive Tutor Algebra*

The Cognitive Tutor Algebra (CTA) is an intelligent tutoring system for high-school Algebra that has been shown to increase student learning by approximately one standard deviation over traditional classroom instruction (Koedinger et al., 1997). It maintains a production-rule model of good and bad problem-solving steps, compares student behaviors to that model, and provides feedback and next-step instruction as appropriate. It also uses knowledge tracing to assess student skills and select problems tailored to individual student needs. The CTA is used in over 2600 classrooms across the United States (www.carnegielearning.com), making it an ideal platform for collecting large amounts of data and conducting externally generalizable research studies.

On one hand, *in vivo* experiments (cf., Koedinger, Aleven, Roll, & Baker, in press) within the classroom setting of the CTA may reduce the chance of reliably extracting a *signal* that the treatment matters from the *noise* of multiple classroom sources of variability. On the other hand, such experiments facilitate the logging of detailed ecological data and patterns that do emerge are likely to be more robust to the variability of real classrooms. Student interface actions are logged as selection-action-input triples, representing the element of the interface with which students interact, the action students took, and the input to the action (Ritter & Koedinger, 1997). For example, entering 25 in a table would be represented as (selection = cell A1, action = enterValue, input = “25”). They also log the correctness of the action (e.g., evaluation = INCORRECT), the skills related to the action inferred using the student model (e.g., “substitution for variable”), and any system feedback and hints a student receives (e.g., “The value you should have entered in the table is 24”). The system also keeps track of the student’s current problem, section, and unit as they progress. Using this problem-solving data, researchers can assess learning as it is occurring, using counts of incorrect attempts or help requests on a particular skill. This type of analysis has led to important results demonstrating the effects of particular modifications to the cognitive tutors, including adding worked examples (Salden, Aleven, Renkl, & Schwonke, 2008) and adding multiple representations (Butcher & Aleven, 2007). We have applied such analysis to evaluating the effects of adding collaboration to intelligent tutoring systems.

2.2. *Reciprocal peer tutoring*

We explored the potential advantages of adding collaboration to the CTA using a reciprocal peer tutoring script. In reciprocal peer tutoring, students of similar abilities take turns tutoring each other on course material. The reciprocal schema is one of the basic schemas proposed by Dillenbourg and Jermann (2008) in their SWISH design model for collaborative learning. Dillenbourg and Jermann argue that the nature of the reciprocal task leads students to interact and construct shared understanding, that is, learn collaboratively. We chose this activity as our research focus for several reasons. We chose this activity as our research focus for several reasons. This type of peer tutoring has been shown to be an effective way of increasing learning in a realistic classroom environment. For example, Fantuzzo, Riggio, Connelly, and Dimeff (1989) found that students in a condition engaging in reciprocal peer tutoring activities learned significantly more than students in both an unstructured collaboration condition and an individual learning condition. Not only did peer tutoring yield increased academic achievement, but it also lowered students' reported distress. Because students in reciprocal peer tutoring take on both the role of tutor and the role of tutee, it is a practical classroom activity, as all students have the chance to benefit from both roles. Further, students in CTA classrooms are very familiar with the concept of being tutored, and it seemed like they would enjoy the role reversal of being able to tutor another student in a similar manner as the CTA.

There are many ways that tutors and tutees benefit from peer tutoring. Roscoe and Chi (2007), in their review of the literature on learning from peer tutoring, concluded that peer tutors benefit from knowledge-building activities, where they reflect on the current state of their knowledge and use it as a basis for constructing new knowledge. During tutoring, peer tutors must monitor their own and their partner's knowledge. If they become aware of gaps in their own knowledge, they will move to repair those gaps, improving their mastery of the domain (Ploetzner, Dillenbourg, Preier, & Traum, 1999). Additionally, peer tutors develop structured networks of knowledge by asking and answering questions and giving and receiving explanations, leading them to make inferences about the subject material and better integrate their knowledge (Roscoe & Chi, 2007). Further, some researchers argue that peer tutors attend more to the domain knowledge that they are to learn because they need to explain it to someone else, and for these reasons, having students *prepare* to tutor can in itself increase learning (Ploetzner et al., 1999; Renkl, 2007). Tutoring in general also has a positive effect on tutee domain learning, particularly at times when the tutee reaches an impasse, is prompted to find and explain the correct step, and is given an explanation if they fail to do so (VanLehn, Siler, Murray, Yamauchi, & Baggett, 2003). However, for these beneficial processes to occur, peer tutors must have sufficient knowledge about the correct solution to the problem to help their partner (Webb, 1989). Otherwise, there may be cognitive consequences (tutees cannot correctly solve problems) and affective consequences (students feel that they are poor tutors and become discouraged; Medway & Baron, 1977). Even though only the tutee solves the complete problem, with the peer tutor acting as a regulator, peer tutoring among students of similar abilities has much in common with other collaborative learning scenarios. The ultimate goal of peer tutoring is for both students to develop a deep understanding of domain concepts, just as in other forms of collaborative learning. To that end, tutees and tutors construct domain knowledge in the process of either solving or explaining problem steps. Additionally, both students take initiative, creating a transactive interaction: The peer tutor determines when to give help by monitoring tutee problem-solving, but the tutee must monitor their own understanding in order to know when to request help or question peer tutor explanations. In reciprocal peer tutoring, where students take turns being tutees and tutors, all students have the opportunity to engage in the same cognitive activities.

As peer tutors do not often show positive tutoring behaviors spontaneously (Roscoe & Chi, 2007), providing the peer tutor with assistance on tutoring skills and domain knowledge can help them to achieve more positive learning outcomes. As described in the introduction, collaborating students are often supported using fixed scripts that outline roles and activities that relate to the desired interactive behaviors. Scripting has also been used successfully in the context of peer tutoring. For example, King, Staffieri, and Adelgais (1998) found that having tutors ask their tutees a series of questions at different levels of depth had a significantly positive effect on tutor learning. Even relatively limiting scripts that leave peer tutors with little freedom in their interactions have had beneficial effects on tutor learning in the classroom (Fantuzzo, King, & Heller, 1992). Another way of increasing the benefits of peer tutoring is to provide students with pre-collaboration training on good tutoring behaviors. Fuchs et al. (1997) trained students to deliver conceptual mathematical explanations and give elaborated help, and showed that their mathematical learning was significantly better than training on elaborated help alone or an individual learning control.

In addition to assessing learning gains by means of posttests, student learning from peer tutoring interactions is often analyzed by collecting tutor and tutee dialogues, using video or audio recording. Dialogues are transcribed, and researchers code the interaction for particular help-giving and help-seeking behaviors. For instance, Webb, Troper, & Fall (1995) have developed an extensive coding scheme for specific and general help requests made by tutees, and levels of elaborated help given by tutors. Similarly, Roscoe and Chi (2007) use a coding scheme that distinguishes between knowledge-telling and knowledge-building behaviors. These types of analyses allow researchers to infer from student dialogue that particular cognitive processes are occurring, link those processes to learning, and link a given intervention to those processes. However, collecting interaction process data in the context of the classroom and in a complex learning setting (e.g., with alternating individual and collaborative phases) is extremely difficult. In addition, these types of dialogue analyses are very time costly and can often only be performed for a small fraction of the process data. Therefore, on top of the potential pedagogical benefits of providing adaptive assistance to peer tutoring, combining intelligent tutoring and peer tutoring might give the researcher access to the rich problem-solving log data common in intelligent tutoring systems, which is not generally recorded in peer tutoring interventions. Using this computer-mediated data would both place the student interaction in context and potentially make it easier to automate parts of the data analysis.

2.3. *Integrating intelligent tutoring and peer tutoring*

One benefit of integrating intelligent tutoring and peer tutoring is the potential learning improvement due to the increased reflective and elaborative demands of the peer tutoring activity. The idea of placing a student in the tutor role in an intelligent tutoring system and providing adaptive support has been explored in the areas of learning companion and simulated student research. Chan and Chou (1997) outlined the space of possibilities for interactions between real learners, real tutors, virtual learners, and virtual tutors, and described two relevant scenarios: One where an agent tutors a human tutoring a human, and one where an agent tutors a human tutoring an agent. They then implemented a distributed reciprocal tutoring system involving two students alternating between learner and tutor roles. Peer tutors were provided with a scaffold, based on a domain model, which helped them to diagnose errors made by the tutee and select a relevant hint. An evaluation of this scenario with five learners showed promising posttest scores. Another "helping the helper" system has been implemented by Vassileva, McCalla, and Greer (2003), where computer agents use peer learner and helper profiles to negotiate tutoring partnerships between students. A further addition to this system provides the helper with more information about the request context, a plan for providing help, and even information about the learner's preferred delivery method (Kumar, McCalla, & Greer, 1999). Finally, people have investigated a human teaching an agent (e.g., Uresti, 2000), or even reciprocal learning scenarios between a human and an agent (e.g., Scott & Reif, 1999), but many of these systems have not implemented adaptive assistance for the peer tutor. One exception is the Betty's Brain system (Leelawong & Biswas, 2008), where a human student tutors "Betty", a computer agent, with the help of another agent "Mr. Davis". This scenario has been found to be effective at promoting learning compared to a traditional intelligent tutoring scenario. Based on these results, there is promise in developing adaptive domain assistance for human-human tutoring for the purpose of improving student interaction and learning.

In parallel, there has been growing interest in the computer-supported collaborative learning community in determining the best way to adaptively support collaborating students (Rummel & Weinburger, 2008) in order to improve on fixed support methods. While the majority of these systems have focused solely on student interaction (see Soller et al., 2005, for review), the most effective adaptive collaborative learning support has leveraged individual domain models in order to augment their collaborative models. Interestingly, these systems tend to be evaluated more thoroughly than other adaptive collaborative learning systems, potentially because individual components of the systems have already been developed and tested. For example, COLLECT-UML, which focuses on UML modeling as the learning domain, uses individual tutoring components to augment its collaborative tutoring components, and has been shown to lead to greater collaborative knowledge over an individual learning system (Baghaei, Mitrovic, & Irwin, 2007). Similarly, CycleTalk combines automated topic detection with a tutorial dialog system that was designed for individual use, in order to adaptively support a collaborative dialog. This system has been shown to be better than fixed support at increasing domain learning (Kumar, Rosé, Wang, Joshi, & Robinson, 2007). Our work involves the analysis of study data drawn from a system constructed using this approach of building upon existing models and systems for individual domain learning.

In particular, our system adds a reciprocal peer tutoring activity to the CTA, and then adaptively supports the peer tutor using the CTA problem-solving models. In an ideal interaction, students being tutored by a peer would benefit from the instruction at least as much as students using the CTA individually, assuming they receive help from the peer tutor at critical problem-solving impasses. Students who are tutoring should benefit further from the additional conceptual demands of reflecting on tutee steps and articulating their reasoning. Thus, one of our hypotheses is that reciprocal peer tutoring using the CTA interface should be better for domain learning than individual use of the CTA. However, because peer tutors are also in the process of learning the domain material, they may not be able to provide the tutee with feedback that is timely or correct. The tutee may then be unable to successfully complete the curriculum problems, and will not benefit from the instruction. Here, a meta-tutor that provides adaptive domain support to the peer tutor during the collaboration has the potential to improve the tutoring quality of the peer, and thus the learning of both students. In traditional peer tutoring activities, domain assistance generally takes the form of preparation on the problems and scaffolding during tutoring (e.g., by giving the tutors the answers to the problems; Fantuzzo et al., 1989). However, adaptive domain support based on an individual learning model, such as that found in the CTA, could form the basis for providing more sophisticated domain assistance to the collaborating partners. Thus, our second hypothesis is that adaptive domain support provided to the peer tutor will improve student interaction, and consequently domain learning, compared to fixed domain support. In the experimental classroom study described in this paper, we evaluate whether the adaptive domain support to peer tutoring is better than fixed domain support to peer tutoring and individual learning using the CTA. It is the combined collaborative dialog and problem-solving data that allows us to thoroughly investigate our hypotheses.

2.4. *Integrating intelligent and peer tutoring data analysis techniques*

In order to get a full picture of the effects of collaboration and adaptive support to collaboration on student learning, we need to be able to link different experimental interventions to student interactions and learning outcomes (as described in Strijbos, Martens, & Jochems, 2004). In our study, where students go through multiple phases of learning (individual and collaborative) and take on multiple roles (tutor and tutee), the data is particularly complex. By integrating intelligent tutoring support with a computer-supported collaborative learning activity, it is possible to view the study data at multiple levels of analysis. While such an in-depth approach was not historically possible in a classroom environment, the combined logging capabilities enabled by intelligent tutoring systems and computer-supported collaborated learning offer us a unique opportunity. Figure 2 displays the different types of data collected by our system. Each level of analysis consists of student interactions with each other and with the system, which is common in computer-mediated collaboration. However, it also consists of the system assessment of the student problem-solving actions, which has been refined particularly in intelligent tutoring system approaches. The highest level of analysis is *activities*, which is a collaborative scripting concept that consists of learning settings (e.g., individual and peer tutoring), phases (e.g., preparation and collaboration) and roles (e.g., peer tutor and peer tutee). Each activity is then composed of several *problems,* which students attempt to solve in sequence. The next level, *attempts*, is influenced by the problem-decomposition approach of intelligent tutoring systems. Each problem is divided into several steps, and students might make several erroneous attempts at a step before completing it correctly. Finally, the lowest level of analysis involves student *interactions*, and is influenced both by collaboration scripts and by intelligent tutoring systems. In order to make an attempt, students interact with each other both in natural language and using the affordances of the interface. These chat and interface actions, and the relationships between them, can be analyzed. Also on this level, the system interacts with the collaborators, providing hints and feedback on the different attempts. The student and system inputs at each level provide insight into what is occurring in the collaboration, and how it might relate to domain learning.

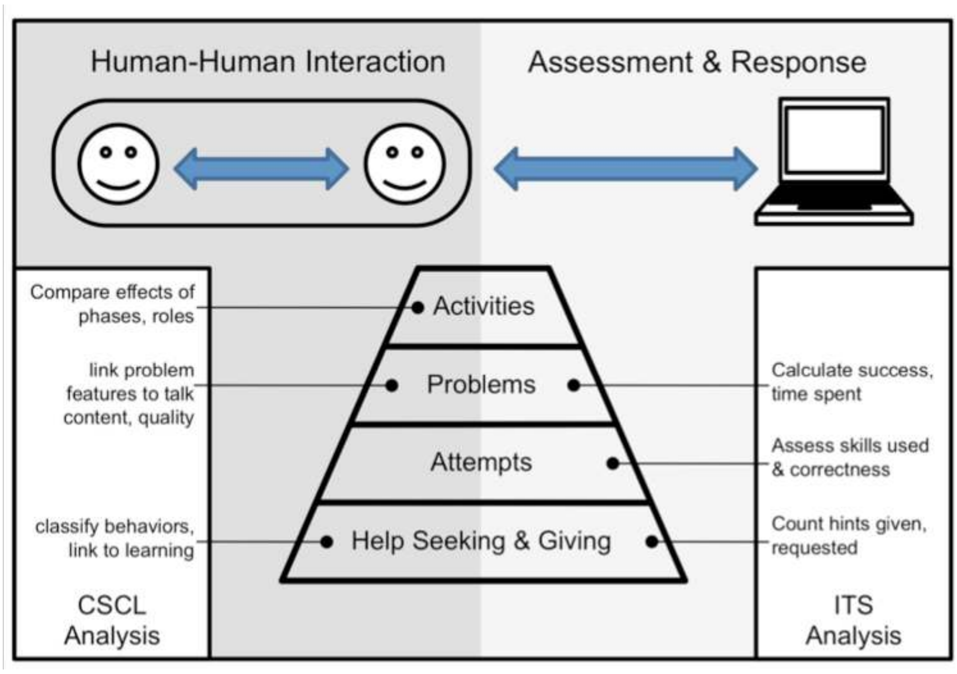

Figure 2. Analysis of interactions possible in ACLS. Interactions consist of both human-human and human-computer exchanges. They can be examined on different levels of granularity, using analyses found in CSCL or ITS approaches.

3. Method

3.1. *Experimental design*

To investigate the effects of collaboration and adaptive support, we conducted an experimental classroom study comparing three conditions: 1. Students used the CTA individually (*individual condition*), 2. Students tutored each other with fixed domain support in addition to the peer tutoring script (*fixed collaboration condition*), and 3. Students tutored each other with adaptive domain support in addition to the peer tutoring script (*adaptive collaboration condition*). Both the adaptive and fixed collaboration conditions included peer tutoring; both the adaptive collaboration condition and individual learning condition included adaptive domain support. We expected the adaptive collaboration condition to learn more than the fixed collaboration condition because of the addition of the domain support, and to learn more than the individual learning condition because of the richness of the peer tutoring activity.

For all conditions, students solved problems from the literal equation solving unit of the CTA, which was identified by classroom teachers as one that was particularly difficult for students to master. In this unit, students are given a prompt like “Solve for x,” and then given an equation like “ax + y = bx + c.” To solve these problems, students must be able to manipulate an equation to move constant (e.g., y) and variable terms (e.g., bx) from one side of an equation to another, factor x, and divide by a coefficient (e.g., a-b). In addition to learning these procedural steps, they must conceptually be able to recognize the difference between constant and variable terms, the difference between positive and negative terms, the distinction between the four major operators on the equation (multiplication, addition, division, subtraction), and the conceptual basis for factoring. Below, we describe each condition in more detail.

3.1.1. *Individual learning condition*

In the individual learning condition, students went through the literal equation solving unit of the CTA exactly as they would during regular classroom instruction. In the CTA, students use the interface displayed in Figure 3 to solve the problems in this unit. They use the menus to select which operation they would like to perform on the equation, and sometimes have to type in the result of the operation as well. They can ask for a hint from the cognitive tutor and receive feedback after errors. Their skills are displayed in the interface in a Skillometer (a visualization of the likelihood that students have mastered the skills required to solve the problems in a given section), and their advancement through the unit relates to the system estimates of their skills. During the study, students proceeded through the six sections that compose this unit. If they finished all the sections before the study was over, they were allowed to move on to other class work.

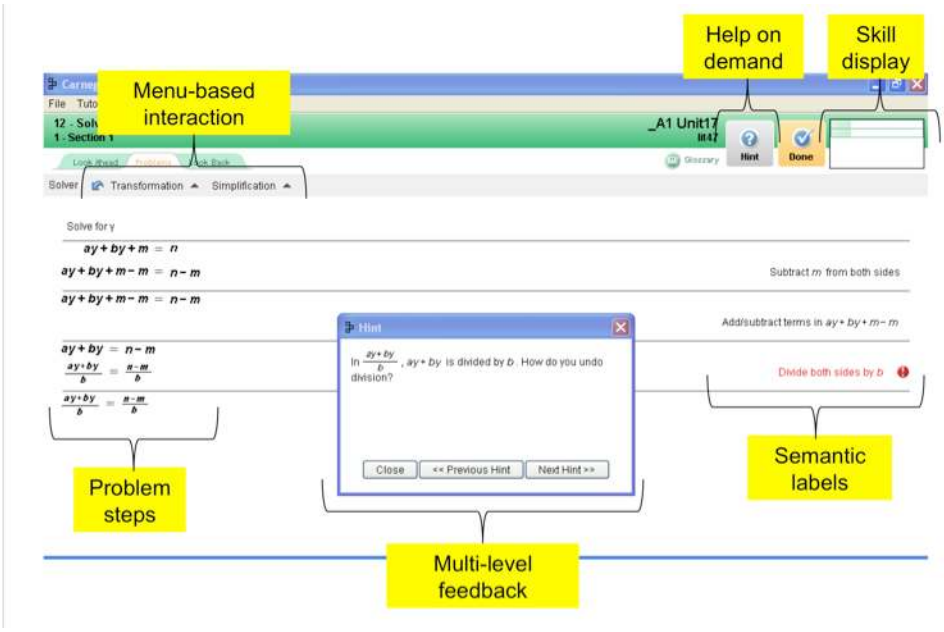

Figure 3. Interface to the CTA in the individual learning condition. As students solve the problem using a menu-based interaction, they receive hints and feedback from the cognitive tutor and information on skills mastered.

3.1.2. *Fixed collaboration condition*

In the fixed collaboration condition, students were placed in pairs, and went through two phases: a preparation phase and a collaboration phase. In the preparation phase, students individually solved the problems they would later tutor (each pair member solved a separate set of problems), using the CTA. Therefore, as in the individual learning condition, students used an equation solver, were given immediate feedback from the CTA when making a mistake, and could ask for a hint from the CTA at any time. After each problem in the preparation phase, we gave students a reflection question to prepare them for tutoring (e.g., "A good question asks why something is done, or what would happen if the problem was solved a certain way. What is a good question to ask about the problem?"). These questions were given on paper, and students did not receive feedback on their answers.

During the collaboration phase, students worked with each other at different computers in the same classroom, taking turns being peer tutors and tutees on alternating problems. Students were seated far apart and discouraged by the classroom teacher from talking to each other out loud. Peer tutees solved the same problems as their tutor had solved in the preparation phase, using the same equation solver tool. However, the tutee could additionally see a chat tool and an enlarged skillometer window. Peer tutors were able to see the peer tutee's actions, but could not solve the problem themselves (see Figure 4). Instead, the peer tutor took the role of the cognitive tutor, marking the peer tutee's actions right or wrong in the solver and adjusting the values of the tutee's skill bars in the skillometer. These activities were designed to trigger the reflective processes that generally lead to peer tutor benefits. In the chat tool, tutees could ask questions,

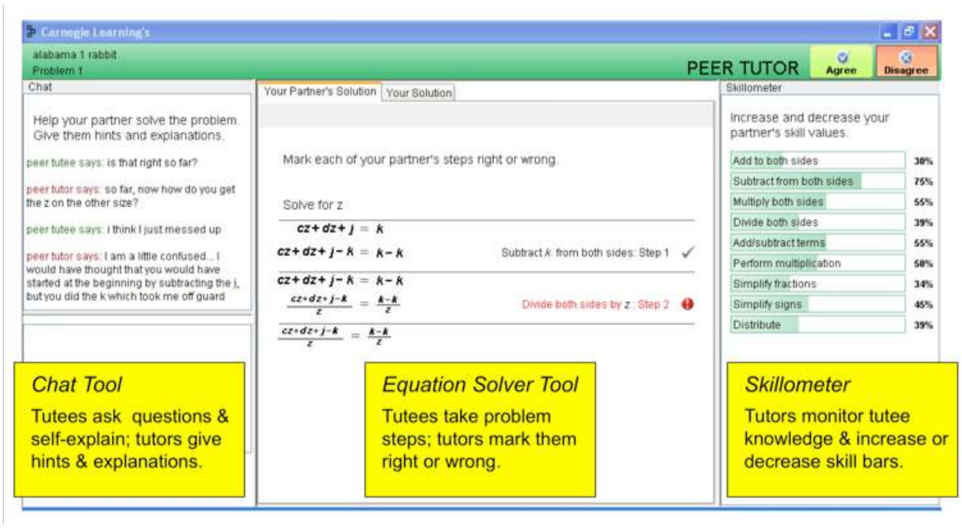

Figure 4. Interface to the peer tutor. Both the peer tutor and tutee see the solver, skillometer, and chat windows. As tutees solve the problem in the equation solver, peer tutors can chat with them, mark their steps, and raise or lower their skill assessments. Peer tutors cannot solve the problem themselves. Tutees can respond to chats in the chat window, and can view the feedback given to them in the solver and skillometer windows.

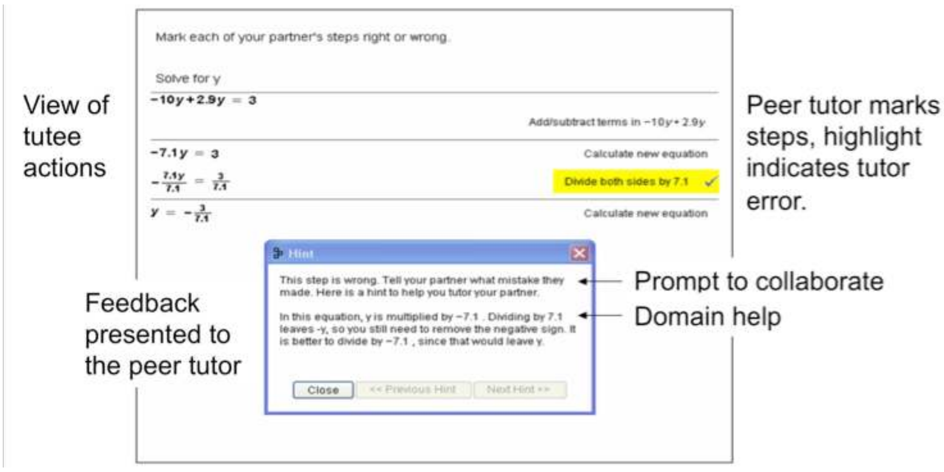

Figure 5. Feedback presented to the peer tutor after the peer tutor marked an incorrect step correct. It consisted of a prompt to collaborate, and the domain help tutees would have originally received.

tutors could give conceptual hints and explanations, and students could follow-up on what had been said. We hypothesized that the potential for elaborated interaction in the chat window would be an improvement over the hint-after-request and feedback-after-incorrect-step mechanisms of the individual CTA. In case peer tutors did not have sufficient expertise to help their tutees, even after the preparation phase, we provided them with fixed cognitive support, consisting of answers to the problem located in a separate tab in the interface (labeled "Your Solution" in Figure 4). Peer tutors could access the tab at any time. Because the support was nonadaptive, if both the tutee and tutor agreed that the problem was finished the students could move to the next problem, even if they had not yet successfully completed the current problem.

3.1.3. *Adaptive collaboration condition*

The adaptive collaboration condition was mostly the same as the fixed collaboration condition, except for the addition of intelligent tutoring support. Peer tutors were given help from the intelligent tutoring system in two cases. First, the peer tutor could request a hint from the CTA and relay it to the tutee. Second, if the peer tutor marked something incorrectly in the interface (e.g., they marked a *wrong* step by the tutee *correct*), the intelligent tutor would highlight the answer in the interface, and present the peer tutor with an error message. Hints and error messages were composed of a prompt to collaborate and the domain help the tutees would have received had they been solving the problem individually (see Figure 5). If both students agreed the problem was done, and were incorrect, the peer tutor would be notified and told to ask for a hint about how to complete the problem. Unlike in the fixed condition, students were not allowed to move to the next problem until the current problem was successfully completed. Messages from the intelligent tutoring system were presented only to the peer tutor, and it was the peer tutor's responsibility to explain them to the tutee. CTA feedback was based on the peer tutor's actions, and not solely on the peer tutee's actions. Therefore, students did not receive CTA feedback at the same rate as students in the individual learning condition. As with the fixed support, peer tutors had access to the problem answers in the interface.

3.2. *Participants*

Participants were 62 high-school students (34 male, 28 female) from five second-year algebra classes at a vocational high school in the United States, taught by the same teacher. There were 10 students in 10th grade, 41 students in 11th grade, and 11 students in 12th grade. Students spent half the day at this high school taking math and various technical subjects (e.g., nursing, electronics). The other half of the day was spent at their "home school" learning conventional subjects. The high school used the individual version of the CTA as part of regular classroom practice. The literal equation solving unit was a review unit for the students, and one that they had already covered in their first algebra class. Based on the assessment of the classroom teacher, the concepts in the unit were difficult for the students to understand, and review was necessary. Students in the collaborative conditions were put in pairs by the classroom teacher, who was told to pair students of similar abilities who would work well together. Because students benefit from being tutors in addition to tutees, and even low-ability students benefit from being placed in the tutor role (see Robinson, Schofield & Steers-Wentzell, 2003, for review), it was important to pair students who felt like they could tutor their partner. Pairing students of similar abilities ensured that students could plausibly function as both tutors and tutees.

Students from each class were randomly assigned to one of the three conditions. Eleven students were excluded from the analysis because either they were absent during a collaborative part of the intervention, or their partner was absent and they could not be re-paired with another student. Another 12 participants did not take the delayed posttest, but were included for all other analyses. The total number of participants included in the analysis was thus 51 for the pre- and posttest (17 students in the adaptive peer tutoring condition, 14 students in the fixed peer tutoring condition, and 20 students in the individual use condition), and 39 students for the delayed posttest (11 in the adaptive peer tutoring condition, 10 in the fixed peer tutoring condition, and 18 in the individual use condition). There were an odd number of students in the adaptive condition because we retained students in the analysis who had an absent partner during an intervention day but were placed with a new partner in the same condition.

3.3. *Procedure*

The study took place over the course of five weeks. Students were given a 15 minute pretest on Monday or Tuesday of the first week, depending on their class schedules. The intervention then took place on two days where students would typically be using the CTA, over two 70 minute class periods. The first intervention day was on Thursday or Friday of the first week, the second was on Thursday or Friday of the following week. On both intervention days, students in the peer tutoring conditions spent half the period in the preparation phase and spent the remaining classroom time taking turns tutoring each other in the collaboration phase. Students switched roles between tutor and tutee after every problem. Students in the individual use condition used the CTA throughout the preparation and collaboration phases. The week after the intervention, students were given a 15 minute posttest. Two weeks later, students were given a 15 minute delayed posttest to assess their long-term retention. On non-intervention days, students continued with their typical algebra curriculum, which involved different units than the literal equation solving unit. This approach was similar to regular classroom practice, where students worked at their own pace while using the CTA, but all received instruction on the same unit during lectures.

3.4. *Measures*

3.4.1. *Outcome measures*

To assess students' individual learning we used counterbalanced pre-, post-, and delayed posttests, each containing 8 questions. The tests were developed by the experimenter and approved by the classroom teacher (see Table 1). The first two questions were scaffolding questions, in that students were either given a problem solution and asked to label each step or given a sequence of step labels and asked to provide the problem solution. The next three questions were parallel to the questions solved during instruction. The final three questions were transfer questions, and asked students to apply their skills in a different context. The questions across the different test versions were parallel but used different numbers and symbols. The tests were administered on paper. We scored answers on the three tests by marking whether the solutions were correct or incorrect. If students got a completely correct solution or reached a nearly correct solution but made a copying error, they received a 1. If students performed one or more conceptual steps incorrectly they received a 0. Points on all the questions were summed. We computed normalized gain scores (Hake, 1998) between the pre- and post-tests and pre- and delayed tests by using the formula gain = (post-pre)/(1-pre). If posttest scores were lower than pretest scores, we used the formula (post-pre)/pre.

3.4.2. *Problem-solving process measures*

In order to analyze student collaborative process, we logged all tutee actions, peer tutor actions, and intelligent tutor responses. We computed the number of problems solved by students in each phase on each study day, the amount of time it took to solve each problem, and whether each problem was successfully completed, unsuccessfully completed (only possible in the fixed condition), or interrupted by the end of the classroom period. Then, for each problem, we computed the number of correct and incorrect problem-solving steps students took. For each step, we computed the number of hint requests to the cognitive tutor that students made, and the amount of feedback from the cognitive tutor that students received. We also calculated the number of tutoring-related interface actions: such as the number of times peer tutors marked a step right or wrong (and whether they were correct in their assessment), and the number of times they consulted the problem answers. All these metrics are data typically available in intelligent tutoring systems.

Table 1. Examples of the three question types used as learning measures. The scaffolding example is a subset of the actual question, which asked the student to label several steps in sequence and provided a sample label.

Question Type	Example
Scaffolding	Label the following step: $az + bc = cz \rightarrow az = cz - bc$
Parallel to Instruction	Solve for z: $az + bc = cz + df$
Transfer	Solve for z: $z^2 - a^2 = c^2 - 2cb$

3.4.3. *Collaborative dialogue coding*

Next, we adapted an approach widely used in collaborative learning research and classified all tutee and tutor chat actions. In general, we segmented the dialog by chat messages, creating a new segment every time students hit enter. However, consecutive lines of chat where the student was uninterrupted by another interface action were classified as the same segment (e.g., a student typed “do you need” and then immediately typed “help”, with no other action being logged between the two chat actions). The experimenter and a second trained rater then independently coded the chat dialogs on two dimensions: help-seeking behavior (Cohen’s kappa = 0.80) and help-giving behavior (Cohen’s kappa = 0.86). The coders trained on 20% of the data and agreement was computed on the remaining 80%. Disagreements on all data were resolved through discussion. The different dimensions are described below.

Our first step was to categorize tutee help-seeking behavior. While in the individual learning condition students could click a hint button to request help, in the collaborative condition students had to make verbal requests to the peer tutor. For our coding, we adapted the coding scheme by Webb, Troper, and Fall (1995), who coded help requests as any statement that was a request for help or indicated confusion. Our data did include direct *requests*, where it was clear that the tutee was expecting an immediate response, often because a question was posed or help was demanded (see Table 2 for examples of all codes). However, tutees also made several *problem-related* statements, where the tutee was not demanding a response from the tutor, but where an on-topic response would be appropriate, such self-explanations or statements of confusion. All other tutee statements were divided into *activity-related* and *off-topic* categories, depending on whether or not they related to the collaborative activity. Next, we defined *help given*, expanding on Webb’s definition of elaborated and unelaborated help (Webb, Troper, & Fall, 1995). Webb divided help received into several degrees of elaboration, ranging from a fully labeled verbal explanation to simply delivering the answer. While these levels mapped to our data, we chose to simply label these forms of help as unelaborated or elaborated, because from a preliminary inspection students either tended to give straightforward instructions or more complex tutoring advice. We also coded hints, where peer tutors provided an explanation for the problem step but did not directly instruct the tutees on what to do, as elaborated help. Our categorization of tutor utterances had five codes: elaborated help, unelaborated help, feedback, activity-related, and off-topic (see Table 2).

Table 2. Coding scheme for tutor and tutee dialogue. We used two codes that related to both students, two additional tutee-specific codes, and three additional tutor-specific codes.

Role	Category	Description	Examples
Tutee	Request	Statement relating to the problem that requires a response from the tutor	"how do I get b by itself", "help"
Tutee	Problem-related statement	Tutee statements containing problem-related content	"so I get w on one side", "I'm lost"
Tutor	Elaborated help	Explanation of a step, hint on how to complete a step, describing an error	"now get m by itself"
Tutor	Unelaborated help	Direct instruction on how to complete all or part of the next step	"factor out t", "then divide"
Tutor	Feedback	Indication of whether a step was right or wrong	"good", "no"
Both	Activity-related statement	Coordination and activity-related statements	"what are you doing?"
Both	Off-topic	Statements not related to the problem or activity	"He's dating her"

4. Results

We began by evaluating our primary hypothesis that the adaptive support condition is better for student domain learning than the fixed support condition and individual learning condition. We then looked at the process data on each level discussed in the above section, moving toward finer and finer granularity. We analyzed the data by individual, so that a given student's actions can be linked to his or her own learning gains and his or her partner's learning gains. For example, the number of errors committed by a student while in the tutee role can be correlated with learning, but so can the number of errors viewed by a student while in the tutor role. At the end of this analysis, we should have a complete picture of the similarities and differences between the paths to learning in each condition.

4.1. *Domain Learning*

We conducted a two-way (condition x test-time) repeated-measures ANOVA, with test-time (pretest, posttest, or delayed test) as the repeated measure. There was a significant effect for test-time ($F[2,72] = 41.303$, $p < 0.001$), but there were no significant differences between conditions ($F[2,36] = 0.881$, $p = 0.423$), and no interaction ($F[2,36]$

= 0.859, p = 0.432). A priori contrasts revealed that the effect was due to the difference between the pretest and the other two tests ($t[36] = 69.541$, $p < 0.001$) and not due to the difference between the posttest and the delayed test ($t[36] = 2.544$, $p = 0.119$). Thus, the different conditions did not have different effects on delayed or immediate learning, and overall students did not show differences between the delayed and immediate measures. For the correlational analyses in this paper described in the following sections, we use the student gain scores between the pretest and posttest and pretest and delayed test, computed as described in section 3.4.1. Table 3 contains the absolute scores of the students who took all three tests, and Table 4 contains their gain scores. It is interesting to note that pretest scores were near floor, despite students' prior familiarity with the unit.

Table 3. Absolute scores on pretest, posttest, and delayed test. Each test had a maximum score of 8.

	Pretest		**Posttest**		**Delayed Posttest**	
Condition	M	SD	M	SD	M	SD
Individual	1.28	1.60	3.00	1.75	3.67	1.78
Fixed	0.90	0.88	3.50	2.17	3.60	2.17
Adaptive	0.82	1.08	2.36	1.57	2.82	1.78

Table 4. Normalized gain scores between pretest and posttest and pretest and delayed test.

	Pre-Post Gains		**Pre-Delayed Gains**	
Condition	M	SD	M	SD
Individual	0.20	0.38	0.26	0.50
Fixed	0.37	0.30	0.30	0.51
Adaptive	0.82	0.40	0.29	0.19

4.2. *Problems*

Our next level of analysis involved the number of problems completed per hour by each condition during the intervention. Because students learned equal amounts across the three conditions, one might expect that the problem-solving rate of each condition would be similar. However, students working collaboratively tend to solve problems slower than students working individually. We further expected the fixed condition to solve fewer problems successfully than the adaptive condition, since it had less relevant domain support. Problems solved may have an impact on the immediate posttest, but is less likely to relate to long-term retention, which is a sign of deeper learning.

We conducted a one-way (condition: individual, fixed, adaptive) ANOVA on the number of problems successfully completed per hour in the collaboration phase of the study (which, for individual learners, was simply the second half of the period). For this particular analysis, we grouped the students in the collaborative conditions by dyad, as the number of problems that one pair member completes is dependent on the number of problems the other pair member completes. Condition was indeed significantly related to problems solved ($F[2,34] = 8.76$, $p = 0.001$), where the adaptive collaboration condition ($M = 17.7$, $SD = 6.69$) and fixed collaboration condition ($M = 13.3$, $SD = 7.71$) solved fewer problems per hour than the individual conditions ($M = 47.0$, $SD = 30.2$). However, there were no differences between the fixed and adaptive conditions. In order to determine if problems completed were related to learning, we correlated total problems *successfully* completed per hour by each student as a tutee with their posttest and delayed test gain scores. Indeed, across all conditions, problems successfully completed per hour were marginally correlated with student learning on the posttest ($r[49] = 0.233$, $p = 0.100$), but not on the delayed test ($r[37] = 0.020$, p = 0.906).

Looking more closely at the collaborative conditions, we can see that differences in the design of the two conditions also led to differences in the number of problems *unsuccessfully* completed. In the fixed condition, students were able to move to the next problem when they thought they were done, regardless of whether they were actually done. Tutees claimed that they were done, and tutors agreed, a mean of 2.50 times ($SD = 1.61$). The mean percentage of times that this exchange occurred out of the number of total problems seen ($M = 8.00\%$, $SD = 2.63\%$) was negatively correlated with the immediate learning gains of the tutee ($r[12] = -0.597$, $p = -0.024$) and the delayed learning gains of the tutee ($r[8] = -0.714$, $p = 0.020$). It was also negatively correlated with the delayed learning gains of the tutor ($r[7] = -0.686$, $p = 0.040$), but not the immediate learning gains of the tutor ($r[10] = -0.214$, $p = 0.504$). In the adaptive collaboration condition, the counterpart of incorrectly moving to the next problem would be the tutee attempting to move to the next problem, the tutor agreeing, and then both being blocked from doing so by the system. Students acting as tutees faced this situation a mean of 2.18 times ($SD = 2.56$). The percentage of times tutees witnessed this exchange out of total problems seen ($M = 5.00\%$, $SD = 2.09\%$) was negatively correlated with learning gains on the delayed posttest ($r[9] = -0.667$, $p = 0.025$), but not on the immediate posttest ($r[15] = -0.007$, $p = 0.980$). Surprisingly, being the tutor during this exchange was positively correlated with learning gains on the delayed posttest ($r[9] = 0.652$, $p = .030$), but not on the immediate posttest ($r[15] = 0.280$, $p = 0.275$). It would seem that being faced with these impasses in the adaptive condition led peer tutors to reflect more on how to overcome them and move to the next problem, an opportunity that they did not have in the fixed condition.

In summary, we found that progress as tutee was correlated with learning on the posttest but not on the delayed posttest. Further, moving on without solving the previous problem was negatively related to learning on the delayed test. On the other hand, witnessing one's tutee getting blocked from moving on, which was only possible in the adaptive condition, was correlated with peer tutor's learning gains. While struggling with the problem may have been detrimental to tutees, viewing this process may have been beneficial for tutors. It may be critical for tutor learning that tutees reach these problem-solving impasses. In the following section, we further explore the relationships between student progress, impasses, and learning.

4.3. *Problem-solving steps*

One explanation for the difference in problems solved was that students struggled with the problems more in the collaborative conditions than in the individual condition, because they did not have the same level of support from the intelligent system. Further, it seems that students in the fixed support condition might commit more errors than students in the adaptive support condition, again due to a lack of sufficient domain assistance. To investigate this hypothesis, we looked at the average number of errors (or incorrect attempts) students made per problem during the collaboration phase. We conducted a one-way (condition: individual, fixed, adaptive) ANOVA, with pretest as a covariate. Pretest was significantly predictive of errors per problem ($F[1,47] = 5.41$, $p = 0.025$), but there were no significant effects of condition ($F[2,47] = 1.738$, $p = 0.187$). The number of errors made by students in the fixed and adaptive peer tutoring conditions were not significantly different from errors made by students working alone (see Table 5, Row 1). We then looked at how the errors made related to learning. As each error was a learning opportunity, we focused on the total error counts, rather than the per problem average. Total errors made were not related to gains on the immediate posttest or delayed test. Viewing errors as a tutor was also not correlated with learning gains on the immediate posttest. However, viewing errors was positively correlated with delayed learning gains (although non-significantly). It appeared that viewing errors related to learning from tutoring, just as observing your tutee unable to proceed to the next problem related to learning from tutoring. These two correlations put together suggest that peer tutors are indeed benefiting from the reflective aspects of tutoring: viewing impasses and considering what might be necessary to overcome them.

Table 5. Frequencies of student progress variables and correlations with learning.

#	Type	Frequencies / problem			Learning gains from being tutored		Learning gains from tutoring	
		Individual	Fixed	Adaptive	Posttest	Delayed Test	Posttest	Delayed Test
1	Errors	M = 1.46 SD = 1.26	M = 1.81 SD = 1.04	M = 2.46 SD = 1.87	$r(49) = -0.113$ $p = 0.432$	$r(37) = -0.063$ $p = 0.702$	$r(27) = -0.058$ $p = 0.763$	$r(18) = 0.354$ $p = 0.126$
2	Help Requests	M = 0.65 SD = 0.81	M = 0.66 SD = 0.52	M = 1.18 SD = 0.52	$r(49) = -0.208$ $p = 0.144$	$r(37) = -0.133$ $p = 0.418$	$r(27) = -0.175$ $p = 0.364$	$r(18) = 0.429$ $p = 0.059$
3	Yes-No Feedback	N/A	M = 0.38 SD = 0.27	M = 0.25 SD = 0.40	$r(27) = 0.153$ $p = 0.427$	$r(18) = 0.051$ $p = 0.832$	$r(29) = 0.353$ $p = 0.052$	$r(19) = 0.454$ $p = 0.038$

4.4. *Helping Behaviors*

Our next level of analysis involved the interaction between the tutee, the peer tutor, and the tutoring system. First, we looked at tutee help-seeking behaviors. Active help-seekers may have been better learners because they were more likely to receive help when it was most appropriate. Additionally, as errors made were related to learning from tutoring, it is possible that tutee help-seeking actions were also related to learning from tutoring. We only used hint requests from the individual condition which occurred during the time period of the collaboration phase (see Table 5, Row 2). We conducted a one-way (condition: individual, fixed, adaptive) ANOVA on hints requested per problem, and found that the number of hints requested in the individual condition was not significantly different from the number of hints requested in chat in each collaborative condition ($F[2,50] = 1.68$, $p = 0.198$). We then determined if we could link making and receiving hint requests to learning gains. Making hint requests was not correlated with immediate or delayed learning gains. Receiving hint requests as a tutor, while not correlated with immediate posttest gains, was marginally correlated with delayed posttest gains ($r[18] = 0.429$, $p = 0.059$). Perhaps the help requests prompted the same reflective processes in the peer tutor as viewing impasses.

In the individual condition, the kind of help given to tutees did not vary, but in the collaborative conditions, the peer tutor chose what kind of help to give, and when to give it. First, we examined the role that verbal yes-no feedback played in the student interaction (Table 5, Row 3). We conducted a one-way (condition: individual, fixed, adaptive) ANOVA on yes-no feedback per problem, and found no significant differences between conditions ($F[1,29] = 0.925$, $p = 0.334$). Feedback given by the tutor was marginally correlated with learning gains as a tutor on the immediate posttest, and significantly correlated with learning gains as a tutor on the delayed test. Feedback received by the tutee was not related to tutee learning gains. Here, because the peer tutor was simply providing yes or no responses, it is not likely that it was the content of the responses that related to learning benefits, but rather the reflective processes that led them to produce the responses. In general, responses with better content were not directly related to learning gains. For example, giving elaborated help was not predicitive of gains on the delayed test for tutors ($r[19] = 0.191$, $p = 0.407$), nor was receiving elaborated help for tutees ($r[18] = 0.108$, $p = 0.649$).

Although giving and receiving elaborated help was not important in isolation, it may be that the quality of the help interacted with the tutee's need for help in order to produce learning gains. In the individual condition, students always received help or feedback after an error or help request, but in the collaborative conditions, that may not be the case. Here, we examine the two extreme examples of the quality and timing of help. First, giving elaborated help after a help request is likely an extremely productive behavior: The tutee needs the help, and using the elaborated help given, the tutee should be able to overcome his or her impasse and complete the next problem step. Table 6 displays the percent requests that tutees made that were answered by the peer tutor with elaborated help, out of total requests made. While this value was not significantly different between conditions ($F[1,29] = 0.136$, $p = 0.715$), answering requests with elaborated help was significantly correlated with learning as a tutor, unlike overall instances of elaborated help. However, as the tutee, having requests answered was not correlated with learning gains. On the other hand, tutees are unlikely to need help immediately after making a correct step, and in particular, they do not need an unelaborated instruction on how to complete the next step. This help is unlikely to be beneficial, and may in fact hinder tutees by preventing them from reflecting on the next step. While the percent unelaborated help when not needed was also not significantly different between conditions ($F[1,27] = 0.011$, $p = 0.918$), it was significantly negatively correlated with tutee delayed learning. To summarize: Giving good help when needed was indeed positively related to tutor learning, while receiving poor help when not needed was negatively related to tutee learning. Interestingly, it is unclear which features of help had a *positive* effect on tutee learning, potentially because of the rareness of good help.

Table 6. Percent good help given when needed and bad help given when not needed. Frequencies and correlations with learning.

#	Type	**Frequencies / problem**			**Learning gains from being tutored**		**Learning gains from tutoring**	
		Individual	Fixed	Adaptive	Posttest	Delayed Test	Posttest	Delayed Test
1	Good help when needed	N/A	M = 31.2 SD = 21.2	M = 23.6 SD = 26.7	$r(27) = 0.301$ $p = 0.113$	$r(18) = 0.267$ $p = 0.255$	$r(27) = 0.398$ $p = 0.033$	$r(18) = 0.476$ $p = 0.034$
2	Poor help when not needed	N/A	M = 12.6 SD = 15.7	M = 13.1 SD = 11.6	$r(27) = -0.331$ p = 0.079	$r(18) = -0.521$ p = 0.019	$r(27) = -0.309$ p = 0.103	$r(18) = -0.055$ p = 0.817

4.5. *Effects of domain support*

We conducted a more exploratory comparison of the effects of cognitive support on learning, using only the adaptive condition. The adaptive condition had both fixed and adaptive feedback available, and thus we could conduct a finer-grained examination of the uses of both types of support. Out of 17 tutors in the adaptive condition, 12 received hints and error feedback from the computer. The other 5 did not ask for hints or mark the problem steps of their tutees, focusing instead on chat communication. Out of an average of 3.50 instances of CTA help ($SD = 3.15$), tutors communicated the help to the tutee 63% of the time ($SD = 42\%$). In general, percent feedback communicated was positively correlated with *tutee* learning gains on the delayed posttest ($r[10] = 0.852$, $p = 0.015$) for the 12 students that used the adaptive feedback. When feedback from the intelligent system was not communicated to tutees, it appeared to lead to damaging confusion on the part of the tutee. Similarly to the adaptive feedback, 13 students used the fixed feedback provided. These students viewed the problem solution a mean of 8.38 times ($SD = 7.96$), over twice the amount of time students received adaptive assistance. 45.8% of the fixed assistance accessed by the tutor was communicated to the tutee ($SD = 32.2\%$), but the percent fixed assistance communicated from the tutor was not correlated with learning from being the tutee. However, communicating fixed assistance was correlated with *tutor* learning gains on the delayed posttest ($r[11] = 0.683$, $p = 0.062$), suggesting that when students actively processed the problem answers they benefitted (or that the good students were more likely to actively use them).

4.6. *Integrating results across data sources*

As a final step, we conducted two regression analyses to better compare the abilities of the variables discussed to predict student delayed learning. We focus here on delayed learning because it indicates long-term retention, and thus is likely a better indicator of deep learning than the immediate posttest. The inferential statistics on the regression results should be taken as suggestive, not conclusive, both because of the small sample (Tabachnick and Fidell, 1996) and because of the correlational nature of the data. Nevertheless, the results can be used to generate causal hypotheses that can then be tested with further experimental manipulation.

First, we constructed a model to predict domain learning using the three variables common to *tutees* in all conditions: problems completed per hour, errors made, and help requested. We further included whether the learning was individual or collaborative as a dummy coded variable (condition; individual = 0, collaborative = 1), and added the interaction terms between condition and all the other variables, as the individual condition differed from the collaborative conditions in several ways. As a whole, the model explained roughly 30% of the variance in delayed gain ($R^2 = .299$, $F[7,38] = 1.891$, $p = 0.105$). Four variables significantly predicted delayed gain: errors made ($\beta = 0.620$, $t[38] = 2.281$, $p = 0.030$), hints requested ($\beta = -0.465$, $t(38) = -2.104$, $p = 0.044$), condition by errors made ($\beta = -1.134$, $t[38] = -3.191$, $p = 0.003$), and condition by hints requested ($\beta = 0.730$, $t[38] = 2.466$, $p = 0.019$). While these results are correlational, it appears that errors made were positively related to delayed gain, but hints requested were negatively related to delayed gain. Interestingly, from the direction of the interaction coefficients it appeared that it is better to try steps in the individual condition than in the collaborative conditions, but better to ask for a hint in the collaborative conditions than in the individual conditions.

Next, we conducted a second regression analysis to predict delayed learning in the two collaborative conditions. We included all the variables that were somewhat correlated with delayed learning and were found in both conditions: errors viewed, help requests received, feedback given, elaborated help given when needed, and unelaborated help received when not needed. We also included errors made and help requests made, as those were significantly predictive of learning in the first regression. The model accounted for a significant proportion of the variance in the delayed gain ($R^2 = 0.783$, $F[7,19] = 6.190$, p = 0.003), although due to the small sample size it is likely that this value is inflated (Tabachnick & Fidell, 1996). Table 7 contains the beta values, t statistics, and p-values for each variable. Elaborated help given when needed was the only variable that was not significantly predictive of delayed gains as a tutor. The variable that was most significantly predictive of delayed gains as a tutor was the yes-no feedback given. Again, these results are correlational, but two interesting elements stand out from this analysis. First, given the postive relationship between learning gains and errors viewed, requests received, and feedback given, students appeared to benefit more from the reflective aspects of tutoring than the articulation of their thoughts. Second, based on the negative relationships between errors made, unelaborated help when not needed, and learning gains, in general tutees may have not received the support they needed to overcome the problem-solving impasses they encountered. In the following section, we will discuss the implications of these results with respect to which aspects of peer tutoring might most benefit from the introduction of adaptive support.

Table 7. Regression results used to predict student delayed learning in collaborative conditions.

Variable	β	***t*(19)**	***p***
Errors made	-0.354	-2.225	0.046
Errors viewed	0.365	2.336	0.038
Requests made	0.353	2.245	0.044
Requests received	0.332	2.091	0.059
Feedback given	0.647	4.403	0.001
Elaborated help given when needed	0.262	1.676	0.120
Unelaborated help received after correct step	-0.375	-2.203	0.048

5. Discussion

While we had hypothesized that the adaptive support condition would be better than the fixed support condition and individual use conditions at increasing domain learning, we found that students learned equally across all conditions. However, using both collaborative dialog and problem-solving data we were able to see differences in the effects each condition had on the path students took to learning. Further, we could take a deeper look at which aspects of student interaction related to learning outcomes. This increased information about how students learned then can be used to inform future designs for adaptive support.

Although learning gains between the individual and collaborative conditions were parallel, students in the two different types of conditions took different paths to learning. It took students in the collaborative conditions far fewer problems to achieve the same learning gains than students in the individual condition (although an equivalent amount of time). This result is in line with other collaborative results that suggest that learning in collaborative conditions is more efficient than learning individually (e.g., Diziol et al., 2008). In domains where problem-authoring is difficult, collaborative conditions may require fewer problems to be designed in order to facilitate student learning. On the other hand, it is possible that had we controlled for the number of problems solved and not for time, students in the individual conditions would have learned as much as students in the collaborative conditions in a shorter amount of time. Further, other than the stark differences between the problems completed in each condition, the individual and collaborative conditions were remarkably similar on the surface. Students made parallel numbers of errors and asked for help at the same rate.

Against our initial prediction, the adaptive and fixed collaboration conditions led to similar domain learning gains. However, each collaborative condition had particular design elements that had unique effects on student interaction. For example, preventing students from moving to the next problem until the previous problem was complete in the adaptive condition may have had an indirectly beneficial effect on tutor learning by leading them to reflect on their misconceptions at these critical moments. Allowing students to move to the next problem without finishing the previous problem appeared to be a design flaw in the fixed condition, as it did not give students the opportunity to reach these beneficial impasses. Even though peer tutors appeared to benefit from adaptive feedback given by the cognitive tutor, not forcing them to communicate it to their tutees may also have been a design flaw, as this event was negatively correlated with tutee learning. Surprisingly, communicating fixed feedback was not related to benefits for the peer tutee, but was related to benefits for the peer tutor. It may be important to give tutors access to materials that they can use to construct conceptual elaborated explanations, and future designs should encourage this behavior.

The problem-solving and collaborative dialog data collected in each condition gave us insight into how students benefitted from being tutors and tutees across both collaborative conditions. Viewing errors, fielding help requests, and giving feedback were all correlated with tutor delayed learning, suggesting the tutors benefitted from the reflective processes triggered by tutee problem-solving actions. The evidence supporting the theory that tutors benefitted from constructing help was more mixed. Although learning was related to communicating fixed support and giving good help when needed, tutors did not benefit from giving good help in general or communicating adaptive assistance received from the CTA. It is possible that increased domain learning led to these good tutoring behaviors, rather than the other way around. Roscoe and Chi (2007) hypothesized that while tutors benefit from knowledge-building, they do not benefit from communicating knowledge that they already know, and it is possible that when looking at student elaborated help, we cannot distinguish knowledge-building from knowledge-telling. Additionally, the benefits of being a tutor may have been offset by the disadvantages of being tutored by a peer. It is potentially problematic that the same elements that led tutors in the collaborative conditions to learn (viewing errors and fielding help requests) are elements that signify a lack of tutee knowledge. Further, we did not find many relationships between collaborative process and tutee learning, although we did find some evidence that receiving help when needed related to tutee learning. It is possible that tutors did not exhibit a sufficient number of positive tutoring behaviors to have a noticeable beneficial effect on tutee learning. It is striking that students in the peer tutor role benefitted from the same interactions that related to less learning for students in the tutee role. It may have been the design of the peer tutoring script itself that lead to the lack of differences between the individual and collaborative conditions in the current experiment. These results do not correspond to other collaborative learning experiments that have demonstrated benefits for collaboration (Lou, Abrami, & D'Appolonia, 2001).

Given the above interpretation of the results, one logical conclusion might be to replace the tutee with a simulated student (e.g., Leelawong & Biswas, 2008). Here, the peer tutor (assisted using adaptive support) would be able to engage in the same reflective processes as we observed in our study, but there would be no tutee to suffer from the peer tutor's lack of expert help. Further, with a simulated student the experimenter could carefully structure the interaction to create situations where peer tutors are likely to engage in beneficial behaviors. While this is an entirely valid approach in its own right, the benefits gained by replacing the human tutee might be counterbalanced by the additional constraints placed on the interaction. The process of receiving requests for help and constructing relevant explanations has been shown in several studies to contribute to peer tutor learning (e.g., Webb & Mastergeorge, 2003), and using the agent as a tutee would limit the expressiveness of the dialog to the limitations of the technology. Further, it might be that students are more motivated to engage in reflective and elaborative processes when they are interacting with a classmate than when they are interacting with a computer. For example, the feelings of social responsibility for your partner that can lead to greater learning from reciprocal peer tutoring activities (Fantuzzo et al., 1992) may not be engaged if students are aware they are interacting with a computer. In general, the nature of an interaction between a peer tutor and a simulated student would likely be very different than an interaction between a peer tutor and a peer tutee.

Another design direction that we are currently exploring is to provide the peer tutor with additional support in helping the tutee. It appears that peer tutors already naturally engage in the reflective processes that lead to learning, but may need more support engaging in elaborative processes and giving good help to their tutees. Thus, we first need to support peer tutors in constructing conceptual, elaborated help, by automatically detecting when their help is unelaborated (e.g., "subtract *x"*), and then by providing assistance at this critical moment. It may be necessary to both remind tutors that they should be giving better help and provide them with sufficient scaffolding to ensure they are capable of doing so. Second, we need to support peer tutors at providing relevant help at moments when tutees have reached impasses. It may be necessary to automatically detect these points where tutees need help, determine whether tutors have provided relevant help, and if not, scaffold them in constructing help that targets tutee misconceptions. In this manner, peer tutors may be able to surpass the helping abilities of intelligent tutors; they will be giving tutees help when they need it, but the help might be more tailored the tutee's level of understanding. However, without sufficient support, peer tutors might continue to fall short in providing support to tutees. Of course, assessing the quality of peer tutoring as it is occurring and determining how to provide assistance that students can use to improve their interaction is a difficult goal that will require many iterative steps to achieve. Nevertheless, there is promise in the approach of increasing the effectiveness of intelligent tutoring systems by augmenting them with supported collaborative activities.

6. Conclusion

In this paper, we described a peer tutoring addition to the Cognitive Tutor Algebra (CTA). We evaluated the effects of the system in a classroom setting by comparing it to a peer tutoring condition with no adaptive support and individual learning using the CTA. In evaluating collaborative learning interventions, it is important to relate process data to outcome data, but relevant process data is often challenging to collect in a classroom environment. Our study set up a complex learning setting, spanning individual and collaborative learning, different phases of learning, and different student roles. By combining intelligent tutoring technology with computer-supported collaboration we were able to collect a variety of process data, ranging from the correctness of problem-solving steps to the dialog between the peer tutor and tutee.

Multiple data sources improved our understanding of the benefits of peer tutoring and adaptive assistance. We were able to specifically link tutor gains to problem-solving behaviors that would logically trigger reflection, such as errors, help-requests, and tutor feedback. Further, the most interesting results required data sources to be combined in a single analysis. Help needed (which links tutee problem-solving and tutor help) and assistance communicated (which links cognitive tutor feedback and tutor help) are the two clear examples of this. These empirical results are not common in other work, potentially because this data is rarely available in an integrated form.

In summary, although there were no outcome differences between conditions, we used the integrated data we collected to develop a more complete picture of how students learned in our environment. We explored the process differences and similarities between individual and collaborative learning, and between learning using fixed and adaptive assistance. We found results in line with existing theories of how peer tutors and tutees benefit from tutoring. Additionally, because of the richness of our data, we were able to contribute to the literature in learning by tutoring in two ways: connecting tutee problem-solving actions to learning by tutoring, and assessing whether tutor help was delivered when it was needed. We are also one of the few studies to analyze how a variety of tutoring processes might relate to learning in our environment, rather than focusing in on a single process. None of these analyses would have been possible without integrating data from multiple sources, and we hope to see these techniques applied in other collaborative learning domains.

Acknowledgments

This research is supported by the Pittsburgh Science of Learning Center, NSF Grant #0354420. We thank Bruce McLaren for his valuable contributions during initial stages of the project. Thanks to Jonathan Steinhart, Dale Walters, and Steve Ritter for their support concerning the use of the Carnegie Learning Cognitive Tutor Algebra code, and to Ido Jamar, Kathy Dickensheets, and the teachers from CWCTC for their motivated involvement in the project. Finally, thanks to Carolyn Rosé, Dejana Diziol, Amy Ogan, and Sean Walker for their helpful comments.

References

Aleven, V., & Koedinger, K.R. (2002). An effective metacognitive strategy: Learning by doing and explaining with a computer-based Cognitive Tutor. *Cognitive Science, 26*(2), 147-179.

Baghaei, N., Mitrovic, A., & Irwin, W. (2007). Supporting Collaborative Learning and Problem Solving in a Constraint-based CSCL Environment for UML Class Diagrams. *International Journal of Computer-Supported Collaborative Learning, 2* (2-3), 159-190.

Bloom, B. S. (1984). The 2 sigma problem: The search for methods of group instruction as effective as one-to-one tutoring. *Educational Researcher*, *13*, 3-16.

Butcher, K., & Aleven, V. (2007). Integrating visual and verbal knowledge during classroom learning with computer tutors. In D.S. McNamara & J.G. Trafton (Eds.), *Proceedings of the 29th Annual Cognitive Science Society* (pp. 137-142). Austin, TX: Cognitive Science Society.

Chan, T.-W. & Chou, C.-Y. (1997). Exploring the design of computer supports for reciprocal tutoring. *International Journal of Artificial Intelligence in Education, 8*(1), 1–29.

Dillenbourg, P. (2002). Over-scripting CSCL: The risks of blending collaborative learning with instructional design. In P. A. Kirschner (Ed.), *Three worlds of CSCL. Can we support CSCL* (pp. 61-91). Heerlen: Open Univeriteit Nederland.

Dillenbourg, P., & Jermann, P. (2007). Designing integrative scripts. In F. Fischer, H. Mandl, J. Haake & I. Kollar (Eds.), Scripting computer-supported communication of knowledge - cognitive, computational and educational perspectives (pp. 275-301). New York: Springer.

Diziol, D., Rummel, N., Kahrimanis, G., Guevara, T., Holz, J., Spada, H., Fiotakis, G. (2008). Using contrasting cases to better understand the relationship between students' interactions and their learning outcome. In G. Kanselaar, V. Jonker, P.A. Kirschner, & F. Prins, (Eds.), International perspectives of the learning sciences: Cre8ing a learning world. *Proceedings of the Eighth International Conference of the Learning Sciences (ICLS 2008), Vol 3* (pp. 348-349). International Society of the Learning Sciences, Inc. ISSN 1573-4552.

Fantuzzo, J. W., King, J. A., & Heller, L. R. (1992). Effects of reciprocal peer tutoring on mathematics and school adjustment: A component analysis. *Journal of Educational Psychology, 84*(3), 331-339.

Fantuzzo, J. W., Riggio, R. E., Connelly, S., & Dimeff, L. A. (1989). Effects of reciprocal peer tutoring on academic achievement and psychological adjustment: A component analysis. *Journal of Educational Psychology, 81*(2), 173-177.

Fuchs, L., Fuchs, D., Hamlett, C., Phillips, N., Karns, K., & Dutka, S. (1997). Enhancing students' helping behavior during peer-mediated instruction with conceptual mathematical explanations. *The Elementary School Journal, 97*(3), 223-249.

Gweon, G., Rosé, C., Carey, R. and Zaiss, Z. (2006). Providing support for adaptive scripting in an on-line collaborative learning environment. In R. Grinter, T. Rodden, P. Aoki, E. Cutrell, R. Jeffries, & G. Olson (Eds.), *Proceedings of ACM CHI 2006 Conference on Human Factors in Computing Systems* (pp. 251-260). New Jersey: ACM Press.

Hake, R.R. (1998). Interactive-engagement versus traditional methods: a six-thousand- student survey of mechanics test data for introductory physics courses. American Journal of Physics, 66, 64 – 74.

Johnson, D. W. & Johnson, R. T. (1990). Cooperative learning and achievement. In S. Sharan (Ed.), *Cooperative learning: Theory and research* (pp. 23-37). NY: Praeger.

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.

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

Koedinger, K. R., Aleven, V., Roll, I. & Baker, R. (in press). In vivo experiments on whether supporting metacognition in intelligent tutoring systems yields robust learning. In D. J. Hacker, J. Dunlosky, A. C. Graesser (Eds.) Handbook of Metacognition in Education. Routledge.

Kollar, I., Fischer, F., & Hesse, F. W. (2006). Collaboration scripts - a conceptual analysis. *Educational Psychology Review, 18*(2), 159-185.

Kollar, I., Fischer, F., & Slotta, J. D. (2005). Internal and external collaboration scripts in webbased science learning at schools. In T. Koschmann, D. Suthers, & T.-W. Chan (Eds.), *The next 10 years! Proceedings of the International Conference on Computer Support for Collaborative Learning 2005* (pp. 331-340). Mahwah, NJ: Lawrence Erlbaum Associates.

Kumar, V., McCalla, G., & Greer, J. (1999). Helping the peer helper. Proceedings of the 9th World Conference on Artificial Intelligence in Education (AI-ED '99), LeMans, France, 325-332.

Kumar, R., Rosé, C. P., Wang, Y. C., Joshi, M., Robinson, A. (2007). Tutorial dialogue as adaptive collaborative learning support. In R. Luckin, K. R. Koedinger, & J. Greer (Eds.) *Proceedings of Artificial Intelligence in Education* (pp. 383-390). IOS Press.

Leelawong, K., & Biswas, G. (2008). Designing learning by teaching agents: The Betty's Brain System. *International Journal of Artificial Intelligence in Education, 18*(3), 181–208.

Lou, Y., Abrami, P. C., d'Apollonia S. (2001). Small group and individual learning with technology: A meta-analysis. *Review of Educational Research, 71*(3), 449-521.

Medway, F. & Baron, R. (1977). Locus of control and tutors' instructional style. *Contemporary Educational Psychology, 2*, 298-310.

Mitrovic, A. & Martin, B. (2002). Evaluating the effects of open student models on learning. In: P. de Bra, P. Brusilovsky and R. Conejo (Eds.) *Proc. 2nd Int. Conf on Adaptive Hypermedia and Adaptive Web-based Systems AH 2002*, Malaga Spain, LCNS 2347, pp. 296-305.

Ploetzner, R., Dillenbourg, P., Preier, M., & Traum, D. (1999). Learning by explaining to oneself and to others. In P. Dillenbourg (Ed.), *Collaborative Learning: Cognitive and Computational Approaches* (pp. 103 – 121). Elsevier Science Publishers.

Renkl, A. (2007). Kooperatives Lernen. In W. Schneider & M. Hasselhorn (Hrsg.), *Handbuch Psychologie, Bd. Pädagogische Psychologie* (pp. 84–94). Göttingen. Hogrefe.

Ritter, S. & Koedinger, K. R. (1997). An architecture for plug-in tutoring agents. *Journal of Artificial Intelligence in Education, 7* (3/4), 315-347. Charlottesville, VA: Association for the Advancement of Computing in Education.

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.

Rittle-Johnson, B., & Alibali, M. W. (1999). Conceptual and procedural knowledge of mathematics: Does one lead to the other? *Journal of Educational Psychology, 91*, 175–189.

Robinson, D., Schofield, J., & Steers-Wentzell, K. (2005). Peer and cross-age tutoring in math: Outcomes and their design implications. *Educational Psychology Review, 17*(4), 327-362.

Roll, I., Aleven, V., McLaren, B. M., & Koedinger, K. R (2007). Can help seeking be tutored? Searching for the secret sauce of metacognitive tutoring. In: R. Luckin, K. Koedinger, & J. Greer (eds.), *Proceedings of the 13th International Conference on Artificial Intelligence in Education* AIED 2007, pp. 203-10.

Roscoe, R. D. & Chi, M. (2007). Understanding tutor learning: Knowledge-building and knowledge-telling in peer tutors' explanations and questions, *Review of Educational Research. 77*(4), 534-574.

Rummel, N. & Weinberger, A. (2008). New challenges in CSCL: Towards adaptive script support. In G. Kanselaar, V. Jonker, P.A. Kirschner, & F. Prins, (Eds.), International perspectives of the learning sciences: Cre8ing a learning world. *Proceedings of the Eighth International Conference of the Learning Sciences (ICLS 2008), Vol 3* (pp. 338-345). International Society of the Learning Sciences.

Salden, R., Aleven, V., & Renkl, A., & Schwonke, R. (2008). Worked Examples and Tutored Problem Solving: Redundant or Synergistic Forms of Support? Paper presented at the *30th Annual Meeting of the Cognitive Science Society*, July 23-26. Washington DC, USA.

Scott, L. A., & Reif. F. (1999). Teaching Scientific Thinking Skills: Students and Computers Coaching Each Other. In *Proceedings of AI-ED 99 World Conference on Artificial Intelligence in Education*, Le Mans, France, 285–293.

Soller, A., Martinez, A., Jermann, P., and Mühlenbrock, M. (2005). From mirroring to guiding: A review of state of the art technology for supporting collaborative learning. *International Journal of Artificial Intelligence in Education, 15*, 261–290.

Strijbos , J. W., Martens , R. L., & Jochems, W. M. G. (2004). Designing for interaction: Six steps to designing computer-supported group-based learning. *Computers & Education, 42*(4), 403-424.

Tabachnick, B.G. & Fidell, L.S. (1996). *Using multivariate statistics* (3rd edition). New York: Harper Collins College Publishers.

Teasley, S., & Fischer, F. (2008). Cognitive convergence in collaborative learning. In G. Kanselaar, V. Jonker, P.A. Kirschner, & F. Prins, (Eds.), International perspectives of the learning sciences: Cre8ing a learning world. *Proceedings of the Eighth International Conference of the Learning Sciences (ICLS 2008), Vol 3* (pp. 360-368). International Society of the Learning Sciences, Inc. ISSN 1573-4552.

Uresti, J. A. R. (2000). Should I Teach my Computer Peer? Some Issues in Teaching a Learning Companion. In G. Gauthier, C. Frasson, & K. Vanlehn (Eds*.). Intelligent tutoring Systems. Fifth International Conference, ITS'2000*, Vol. 1839 of Lectures Notes of Computer Science, Springer-Verlag, 103–112.

VanLehn, K. (2006). The behavior of tutoring systems. *International Journal of Artificial Intelligence in Education, 16*(3), 227-265.

VanLehn, K., Siler, S., Murray, C., Yamauchi, T., & Baggett, W. (2003). Why do only some events cause learning during human tutoring? *Cognition and Intruction, 21*(3), 209-249.

Vassileva, J., McCalla, G., and Greer, J. (2003). Multi-Agent Multi-User Modeling in I-Help. *User Modeling and User-Adapted Interaction: The Journal of Personalization Research, 13*, 179-210, DOI: 10.1023/A:1024072706526.

Webb, N. M. (1989). Peer interaction and learning in small groups. *International Journal of Educational Research, 13*, 21–40.

Webb, N. M., & Mastergeorge, A. (2003). Promoting effective helping behavior in peer-directed groups. *International Journal of Educational Research, 39*, 73-97.

Webb, N. M., Troper, J., & Fall, R. (1995) Constructive activity and learning in collaborative small groups. *Journal of Educational Psychology, 87*(3).

Weerasinghe, A., Mitrovic, A. (2006). Facilitating Deep Learning through Self-Explanation in an Open-ended Domain. *Int. J. of Knowledge-based and Intelligent Engineering Systems (KES)*, *10*(1), 3-19.