Computer-Supported Collaborative Learning (2011) 6:279–306
DOI 10.1007/s11412-011-9111-2

Designing automated adaptive support to improve student helping behaviors in a peer tutoring activity

Erin Walker · Nikol Rummel · Kenneth R. Koedinger

Received: 9 July 2010 / Accepted: 10 January 2011 /
Published online: 8 February 2011

Abstract Adaptive collaborative learning support systems analyze student collaboration as it occurs and provide targeted assistance to the collaborators. Too little is known about how to design adaptive support to have a positive effect on interaction and learning. We investigated this problem in a reciprocal peer tutoring scenario, where two students take turns tutoring each other, so that both may benefit from giving help. We used a social design process to generate three principles for adaptive collaboration assistance. Following these principles, we designed adaptive assistance for improving peer tutor help-giving, and deployed it in a classroom, comparing it to traditional fixed support. We found that the assistance improved the conceptual content of help and the use of interface features. We qualitatively examined how each design principle contributed to the effect, finding that peer tutors responded best to assistance that made them feel accountable for help they gave.

Keywords Adaptive collaborative learning support · Adaptive scripting · Reciprocal peer tutoring · Intelligent tutoring · In vivo experimentation

Introduction

Through participation in collaborative activities students socially construct knowledge (Schoenfeld 1992). They can elaborate on their existing knowledge and build new knowledge when they articulate their reasoning (Ploetzner et al. 1999), integrate other group members' reasoning (Stahl 2000), reflect on misconceptions, and work toward a shared understanding (Van den Bossche et al. 2006). However, for collaboration to be

E. Walker (✉) · K. R. Koedinger
Human-Computer Interaction Institute, Carnegie Mellon University, Pittsburgh, PA, USA
e-mail: erinwalk@andrew.cmu.edu

K. R. Koedinger
e-mail: koedinger@cmu.edu

N. Rummel
Institute of Education, Ruhr-Universität Bochum, Bochum, Germany
e-mail: nikol.rummel@rub.de

effective at engaging these processes, students need to display positive collaborative behaviors (Johnson and Johnson 1990), and they generally do not do so without assistance (Lou et al. 2001).

Small-group collaboration can be supported in several ways: through the use of human facilitation to guide the interaction (Hmelo-Silver 2004; Michaels et al. 2008), pre-collaboration training (Prichard et al. 2006; Saab et al. 2007), or scripting of the collaborative interaction by giving students designated roles and activities to follow (Fischer et al. 2007; Kollar et al. 2006). While human facilitation can indeed be effective, it is resource intensive, as it requires an expert facilitator to guide each group's discussion. Training and scripting are less resource intensive, but students may not be capable of or motivated to follow the instructions given (Ritter et al. 2002). In a face-to-face collaboration context, there is no way for these techniques to ensure that they do so. An increase in the presence of computer-mediated collaborative activities in the classroom has changed the way collaboration can be structured, as script elements can be embedded in the interface: Roles can manifest themselves through the types of collaborative actions students can perform using the system, phases can be strictly enforced, and prompts can take the form of sentence classifiers or starters, where students complete open-ended sentences such as "I agree, because..." However, this increase in support comes with a potential decrease in motivation, as this level of support can overstructure collaboration for students who already know how to collaborate (Kollar et al. 2005). Further, students often fail to comply with script elements such as sentence starters (Lazonder et al. 2003), perhaps because they do not know how to use them effectively or are not motivated to do so. For example, if students repeatedly use sentence classifiers to type something off-topic, such as "I agree because... I'm getting hungry," this is unlikely to contribute to a beneficial interaction.

A promising new method for facilitating computer-supported collaborative activities in the classroom is by providing students with *adaptive* collaborative learning support (ACLS). In theory, this approach should be more effective than fixed support alone, as students would always receive a level of assistance appropriate to their collaborative skill, and the intelligent system could verify that students are, in fact, improving their collaboration (Rummel and Weinberger 2008). Studies comparing automated adaptive support to fixed support have indeed been promising (Baghaei et al. 2007, Kumar et al. 2007), but research into ACLS is still at an early stage. In designing ACLS people have mainly adapted individual learning paradigms, such as providing explicit feedback directly to the unproductive collaborator (see Soller et al. 2005, for a review). For example, the system *COLLECT-UML* responds to a lack of elaboration by saying: "You seem to just agree and/or disagree with other members. You may wish to challenge others' ideas and ask for explanation and justification" (Baghaei et al. 2007). This form of feedback has been demonstrated to be successful in individual learning (e.g., Koedinger et al. 1997), as students can immediately reflect on how the feedback applies to their current activity and make appropriate changes to their behavior. However, it may be less appropriate for collaboration, and, in fact, Kumar et al. (2007) found that students tended to ignore adaptive prompts while collaborating. Students may ignore adaptive feedback because it violates Gricean maxims of the conversation (e.g., appears irrelevant to the task; Bernsen et al. 1997) or disrupts the perceived safety of the collaborative context (Nicol and Macfarlane-Dick 2006; Van den Bossche et al. 2006). Two recent studies have demonstrated that socially sensitive features of adaptive support are indeed important for getting positive outcomes from the support (Chaudhuri et al. 2009; Kumar et al. 2010). Given the complex set of options and interactions involved in collaboration support, any null effects found in comparing adaptive to fixed support might be due to limitations in the support design, and not because adaptive support is ineffective per se. It is therefore important to explore the full design space of support possibilities.

This project focuses on how to design adaptive support to improve the quality of collaborative student interaction. We investigate this broader question in the context of a system that we have developed for supporting help-giving behaviors in a reciprocal peer tutoring scenario for algebra. Our overall program of research has leveraged a paradigm evolved from the *in vivo* experiments described by Koedinger et al. (2009). *In vivo* experiments lie at the intersection of psychological experimentation and design-based research, as defined by Collins (1999) and later expanded upon by Barab and Squire (2004). Like psychological experimentation, an *in vivo* experiment involves the manipulation of a single independent variable and the use of fixed procedures to test a set of hypotheses. In addition, like design-based research, an *in vivo* experiment takes place in real-world contexts that involve social interaction, and characterizes the relationships between multiple process variables and outcome variables.

We would argue that for *in vivo* experimentation to be successful it can be helpful to incorporate further elements of design-based research outside those used in a single in vivo experiment: the use of participant co-design and analysis to develop a profile of what is occurring and inform flexible, iterative design revisions (Beyer and Holtzblatt 1997). *Iterated in vivo experimentation,* where we use a design-based research process to create an intervention, deploy the intervention using an *in vivo* experiment, and then interpret the effects through a design-based lens, may be a more effective way of theory building than executing an in vivo experiment in isolation. This concept of iterated *in vivo* experimentation is similar to that of action research (Brydon-Milier et al. 2003), with a few key differences. Under iterated *in vivo* experimentation, theory-building is a driving force in the research agenda, in addition to effecting social change, and the use of both quantitative and qualitative methods are advocated.

In this paper, we discuss our four phase program of iterated *in vivo* experimentation for adaptively supporting the quality of peer tutor help-giving. First, we used a human-computer interaction design method called Speed Dating (Davidoff et al. 2007), which led to the identification of three design principles for adaptive collaboration assistance in this context (*Phase 1: Needs Identification*). Based on these principles, we augmented an existing peer tutoring system with adaptive support, including reflective prompts triggered by elements of the help given by peer tutors (*Phase 2: Assistance Design*). We deployed the system in an *in vivo* experiment, and quantitatively analyzed its effects, comparing the adaptive assistance to parallel fixed assistance for effects on student help-giving (*Phase 3: In Vivo Experiment*). We then returned to a design-based methodology to qualitatively examine how our three design principles more directly influenced the student interactions (*Phase 4: Contrasting Cases*). We conclude this paper by discussing the theoretical and design implications of our results, placing our system in the context of other ACLS systems.

Reciprocal peer tutoring

We explore how to design adaptive support to improve help-giving behaviors among peers. Help-giving is an important part of many collaborative activities, and is a key element of the productive interactions identified by Johnson and Johnson (1990) that contribute to learning from collaboration. In giving help, even novice students benefit; they reflect on their peer's error and then construct a relevant explanation, elaborating on their existing knowledge and generating new knowledge (Roscoe and Chi 2007). Thus, improving help-giving is likely to benefit the help giver. Further, supporting help-giving might have indirect benefits for the help receiver, as students tend to benefit most from receiving help that: arrives when they reach an impasse, allows them to self-explain, and, if necessary, provides an explanation that is conceptual and targets their misconceptions (VanLehn et al. 2003; Webb 1989; Webb and Mastergeorge 2003).

Unfortunately, most students do not exhibit positive helping behaviors spontaneously (Roscoe and Chi 2007). Thus, during collaboration students may fail to help each other well or even at all. Specifically, students are often more inclined to give each other instrumental help (e.g., “subtract *x*”). They rarely provide conceptual, elaborated help that explains why, in addition to what, and references domain concepts (e.g., “subtract *x* to move it to the other side”). This tendency decreases the likelihood that either student engages in elaborative knowledge-construction processes and benefits from the interaction (Webb and Mastergeorge 2003). Promoting the conceptual content of student help has benefits for the interaction as a whole (Fuchs et al. 1997), and is the major focus of our peer tutoring support.

One technique for facilitating student help-giving is by means of employing a reciprocal schema, where first one student is given artificial expertise in a particular domain and is told to regulate the problem solving of a second student, and then the roles are reversed and the second student becomes the expert (Dillenbourg and Jermann 2007). As part of their role, the expert must monitor their partner’s problem solving and offer appropriate help when it is needed. Examples of this class of collaborative activities are dyadic activities such as reciprocal teaching by Palincsar and Brown (1984), mutual peer tutoring by King et al. (1998), and reciprocal peer tutoring by Fantuzzo et al. (1989). Several of these activities have been successful at increasing student learning in classroom environments compared to individual and unstructured controls (Fantuzzo et al. 1989; King et al. 1998; Fuchs et al. 1997). They have been effective for both low and high ability students, but only when further support is provided to students in order to assist them in helping each other effectively. For example, Fuchs et al. 1997 trained students to deliver conceptual elaborated mathematical explanations, and showed that their mathematical learning was significantly better than elaborated help training alone (without specific emphasis on conceptual content).

To explore the potential of adaptive support for help-giving, we have developed a reciprocal peer tutoring environment as an addition to the Cognitive Tutor Algebra (CTA), a successful intelligent tutoring system for individual learning in high-school Algebra (Koedinger et al. 1997). We designed the environment to be used in a scenario where first students prepare to tutor each other on different problems, and then are seated at different computers in the same classroom and alternate being tutors and tutees. The environment encourages students to collaborate in a shared workspace, and talk to each other using a chat window. In this way, it has much in common with the successful Virtual Math Teams project, where groups of students get together to discuss mathematics using a shared workspace and unstructured chat (Stahl 2009). Our system also draws from other adaptive systems that support peer help, such as *IHelp*, where computer agents use peer learner and helper profiles to negotiate tutoring partnerships between students (Vassileva et al. 2003). However, unlike *IHelp,* our system supports peer tutors as they tutor, attempting to improve the conceptual, elaborated help they provide. Thus, the environment also borrows from single-user systems that have students tutor a virtual peer, some of which include an adaptive agent that helps the student be a better tutor (e.g., Leelawong and Biswas 2008). Our research stands out in its extended development of synchronous automated adaptive support of complex human-human interaction, in the context of peer tutoring. In the next section, we discuss Phase 1 of our iterated in vivo experimentation program: our exploration of potential designs for adaptive interaction support for peer tutoring.

Phase 1: Needs identification

The first phase of our work centered on an exploration of potential designs for adaptive interaction support for peer tutoring. We generated several different ideas for adaptive support, and then used the Speed Dating method (Davidoff et al. 2007) to gather student reactions. From this process we derived three principles of ACLS design.

Ideation

Drawing inspiration from existing forms of support for individual and collaborative learning, we generated several ideas for adaptively supporting reciprocal peer tutoring that went beyond the individual learning model of presenting explicit feedback to the collaborator who needs support.

Reflective prompts

One idea for delivering adaptive support to collaborators is to mimic the support that human facilitators provide in face-to-face groups. In accountable talk as described by Michaels et al. (2008), a teacher directs a classroom using several different reflective “moves”, such as asking a student to expand on an utterance (“Why do you think that?”). Instead of presenting a single student with very explicit feedback, it may be beneficial to present all students involved in the interaction with questions that prompt further reflection and reasoning. While other adaptive systems have presented feedback publicly to both users (e.g., Constantino-Gonzalez et al. 2003), it is rare for adaptive systems to pose open-ended reflective prompts. While it is true that there are valid technical reasons for this design decision, the ability of systems to analyze open-ended responses is increasing (McNamara et al. 2007), and looking to the future, it is important to explore the potential of these prompts. Further, posing these prompts at appropriate times may be beneficial for triggering cognitive processes (Chi et al. 1994), even if the system does not follow up on student responses.

Peer-mediated feedback

Some effective fixed collaborative learning scripts attempt to get individual students to elicit certain responses from their partners; for example, by having students ask their partner a series of questions at increasing levels of depth (e.g., King et al. 1998). In our second idea, peer-mediated feedback, the system provides interaction guidance to the partner of the student whose behavior we would like to change. For example, if one student is not self-explaining their problem-solving steps, we can prompt their partner: “Did you understand what your partner did? If not, ask them why.” Students who receive the feedback are thus encouraged to self-regulate their learning, prompting them to request the help they need from their partner. For the students whose behavior we would like to change, receiving a prompt from a partner might feel more natural and comprehensible than receiving computer feedback.

Adaptive resources

Adaptive resources, instead of explicitly telling students how to improve their behavior, provide students with resources to help them to make the necessary changes. This approach is drawn from adaptive hypermedia, where the information that is available to students changes in accordance with their knowledge (Brusilovsky 2001). In a fixed support approach developed by Fuchs et al. (1997), students were trained in delivering conceptual mathematical explanations, using an alternating program of video clips and classroom discussion. In an adaptive system, a video related to each collaboration skill could be presented when a student may be thinking of applying the skill (for example, while preparing explanations for a given problem), and additional materials surrounding the video could incorporate specific information about the current problem for collaborating students. While this specific approach has rarely been used in ACLS systems, visualization systems have been developed that mirror back to students' aspects of their collaborative performance (Soller et al. 2005). Augmenting these systems to present more information to students about reaching ideal performance might be a fruitful area of research.

Speed dating process

Our next step was to use these assistance concepts as a basis for exploring user perceptions relating to adaptive support. We applied a design method called Speed Dating (Davidoff et al. 2007), which takes a sketch-based approach to give the designer insight into user needs. The aspect of Speed Dating we leveraged involves the use of focus groups to discuss several potential design scenarios in rapid succession. We sketched 12 scenarios for adaptively supporting a reciprocal peer tutoring activity, based both on the ideas described above and on traditional ACLS. The support sketches varied according to the collaborative situation that triggered the support, with four sketches designed to support peer tutors unsure how to give help, four sketches designed to prevent peer tutors from giving over-enthusiastic help, and four sketches designed to prevent peer tutors from giving simplistic instructions. Each scenario leveraged particular aspects of the ideas described in the previous section. Fig. 1 shows a sample scenario that we presented to students representing peer-mediated feedback: In response to the peer tutor giving unasked-for help, the tutee is told to ask his partner to let him try the step before helping. We then assembled three groups of volunteer high school students with four students per group. Groups were pulled out of class and interviewed at the school. We presented the 12 support sketches to students, and asked for their reactions to each idea.

Accountability and efficacy design principles

Two motivational influences reappeared in student discussions: feelings of accountability for tutee learning, and a desire for tutoring efficacy. Students appeared to take their potential role as peer tutors very seriously, saying when considering a tutoring error: "Maybe he's going to be messed up—I wouldn't want that to happen" (Group 1). They wanted to feel like good tutors and be perceived as good tutors, responding very positively to a scenario where the computer offered public praise in the chat window: "I really like the one where the computer joins in on the IM… You gave that person good advice, both students see it" (Group 1). Based on this analysis, students who do not feel like capable tutors may disengage with the activity or simply give their partners the answer in order to increase their feelings of efficacy.

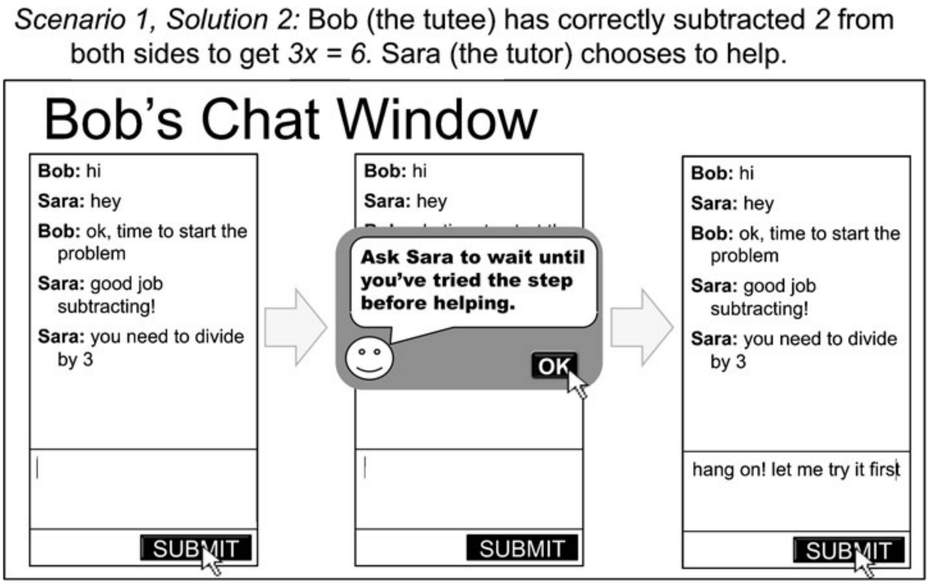

Fig. 1 Speed Dating scenario. In this scenario, the tutee is encouraged to self-regulate their own learning by asking the peer tutor to refrain from helping until the tutee has tried the step. Students were presented with 12 scenarios in rapid succession and asked discussion questions

There are two main implications of these motivational factors with respect to designing assistance provided to peer tutors: First, assistance could be designed to leverage the feelings of accountability already present in tutoring interactions in order to encourage peer tutors to give help in effective ways (*Accountability Design Principle*). For example, presenting interaction feedback and praise publicly in the chat window where both students can see it might encourage peer tutors to apply the advice. Second, it is necessary for assistance in general, and in particular for assistance designed to increase accountability, to avoid threatening peer tutors' beliefs that they are capable tutors, but instead to increase their sense of control over the situation (*Efficacy Design Principle*). Any assistance given by the computer should avoid undermining the peer tutor's control over the interaction, and for this reason, students overwhelmingly rejected the idea of commentary on peer tutor actions being given to the tutee, saying that this was "like your teacher talking over your shoulder" (Group 2). Instead, students preferred assistance that put computers and peer tutors on more equal footing, such as reflective prompts delivered by computers in the chat window ("the computer's asking—I kind of like that… I think the computer should just go ahead and do it in the chat window"—Group 3). By positioning computers and peer tutors as collaborators (see Chan and Chou 1997, for examples of this strategy in individual learning), we may be able to preserve tutoring efficacy, increasing peer tutor motivation to give good help.

Relevance design principle

When exploring student perceptions of different support designs, we also found that students particularly focused on how relevant the help appeared to be to their task, and how little it disrupted their interaction. On a broad level, it was clear that students wanted to get system feedback that they could use ("If it [the computer] says something we needed to know then it would be ok"—Group 2). By and large, students cited cognitive help on how to solve the problems as useful feedback, but surprisingly, what they wanted to receive was not simply a hint targeted at the next problem step. Students said that the adaptive hints were not always very informative ("the hint—doesn't really tell you much"—Group 2), and admitted that therefore they would be likely to take advantage of the hints facility ("You could just be clicking the hint button, to like, get the answers"—Group 3). Instead, students stated that they preferred hints that gave both the high-level concepts relevant to each problem step, and specific illustrations of the concepts. One student even suggested support that "give[s] you an example problem, but explains the steps to you and explains how they get the answer" (Group 3). Despite all this discussion about the usefulness of cognitive feedback, there was nearly no talk about the usefulness of interaction feedback, suggesting that students perceived interaction feedback as less relevant than cognitive feedback. As we consider interaction feedback highly relevant to student learning, this finding may be problematic.

This analysis leads us to recognize the importance of designing adaptive support so that it appears relevant to students (*Relevance Design Principle*) in order to motivate them to incorporate the assistance into their own interactions. As students believe that cognitive support is relevant but do not recognize the relevance of interaction support, it might be that any interaction feedback given to students should be linked to cognitive feedback, to make the interaction feedback more concrete and immediately applicable. Telling students: "You should explain why to do a step in addition to what to do. For example, on the next step your tutee should be trying to isolate the *y*" might make the help seem more relevant than simply just telling them "You should explain why to do a step in addition to what to do." Another technique for making the collaboration support more relevant to student interaction is by clearly linking the support to what peer tutors themselves want to do. For example, help on how to give an explanation would be perceived as maximally relevant when students are actively trying to give their partner an explanation.

Discussion

By generating diverse ideas for support, and then using a needs-validation method called Speed Dating, we generated three design principles for supporting students in collaborating with each other: accountability, efficacy, and relevance. At first glance, these principles may not appear surprising: Effects of accountability and need for efficacy have been documented in previous peer tutoring literature (Fantuzzo et al. 1989; Robinson et al. 2005), and the need for relevance is well-known (Bernsen et al. 1997). However, one of the surprising elements of the Speed Dating analysis was that the computer is ascribed a social rather than functional role when interacting with the peer tutor. This result is not necessarily predicted by the literature, which suggests that in a computer-mediated context, people react differently to humans and computers (Rosé and Torrey 2005). The fact that the computer support in this context might conflict with the peer tutor's role as "teacher", threatening peer tutor feelings of being a good tutor, is interesting and important for the support design. Although previous work discusses how efficacy is important to peer tutors, it is not clear from this literature that efficacy, among many other motivational factors, should be a primary consideration in the design of computer support. In addition, the design exercise further revealed particular implications of the three design principles for accepting or rejecting certain varieties of assistance. For example, students' opinion that feedback to the peer tutor should never be delivered solely to the tutee is an important insight into how manipulating the target of the support might affect feelings of efficacy. It also argues for rejecting peer-mediated feedback delivered to the tutee as an option for assistance. The insights gleaned from the Speed Dating activity formed the basis for our design of adaptive support for peer tutor help giving, which is described in the next section.

Phase 2: Assistance design

In our previous research, we constructed APTA (the Adaptive Peer Tutoring Assistant) to support students in tutoring each other on literal equation solving. As part of this system, we developed adaptive cognitive support to facilitate peer tutor reflection on errors and improve the correctness of peer tutor help (Walker et al. 2009). We used the design principles introduced above and student suggestions during the Speed Dating activity to refine the initial assistance scenarios, yielding three broad forms of assistance:

1. Hints on demand. Given when the peer tutor does not know how to proceed.
2. Adaptive resources. Provided when the peer tutor needs support in constructing help.
3. Reflective prompts. Delivers feedback on help peer tutors have already given.

By incorporating a variety of adaptive help-giving assistance, we could examine how different assistance affected motivation, interaction, and learning. We generated prototypes, and conducted four iterations of think-aloud sessions, creating higher-fidelity versions with each iteration, until we arrived at our current system.

Previous version of APTA

In APTA, students work on literal equation problems where they are given an equation like "$ax + by = c$" and a prompt like, "Solve for x". Students are seated at different computers, and at any given time, one student is the peer tutor and the other is the tutee. Tutees can

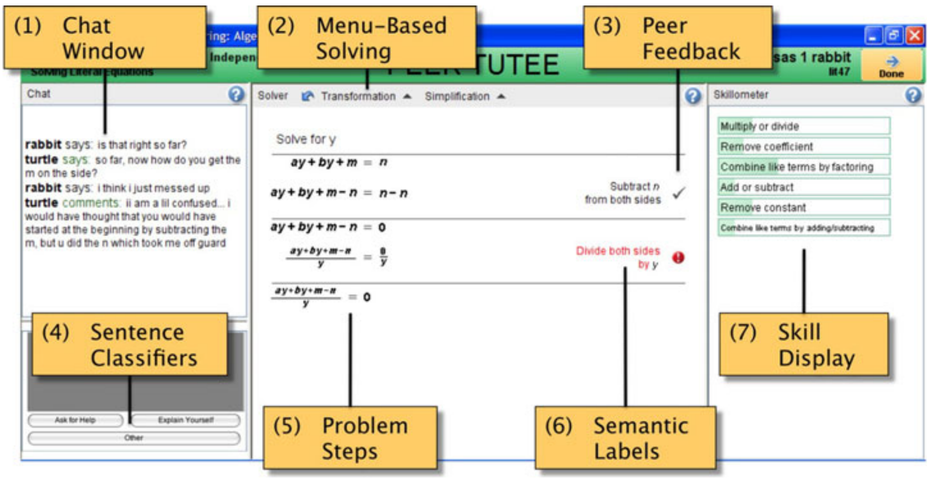

Fig. 2 Tutee's problem-solving interface. The tutee solves problems using the menu, chats with their partner in the chat window, and receives feedback in the solver and skillometer

perform operations on the equation with a menu-based interaction used in the common, individual version of the Cognitive Tutor Algebra (*CTA*). See Fig. 2 for a screenshot of the tutee's interface. Using the menus, students can select operations like "subtract from both sides", and then type in the term they would like to subtract (#2 in Fig. 2). For some problems, the computer then performs the result of the operation and displays it on the screen (#3 and #4 in Fig. 2); for others, the tutee must type in the result of the operation themselves. The peer tutors can see the tutee's actions on their computer screen, but are not able to perform actions in the problem themselves (see Fig. 3 for a screenshot of the peer tutor's interface, #5). Instead, the peer tutor can mark the tutee's actions right or wrong (#6 in Fig. 3), and raise or lower tutee skill assessments in the skillometer window (#1, Fig. 3). Students can discuss the problem in a chat window (#1 in Fig. 2 and #4 in Fig. 3).

To facilitate the discussion in the chat window, we included a common form of fixed scaffolding: sentence classifiers. This form of fixed scaffolding is thought to be pedagogically beneficial by making positive collaborative actions explicit in the interface and encouraging students to consider the type of utterance they wish to make (Weinberger et al. 2005). We asked peer tutors to label their utterances using one of four classifiers: "ask why", "explain why wrong", "give hint", and "explain what next" (#8 in Fig. 3). Students had to select a classifier before they typed in an utterance, but they could also choose to click a neutral classifier ("other"). For example, if students wanted to give a

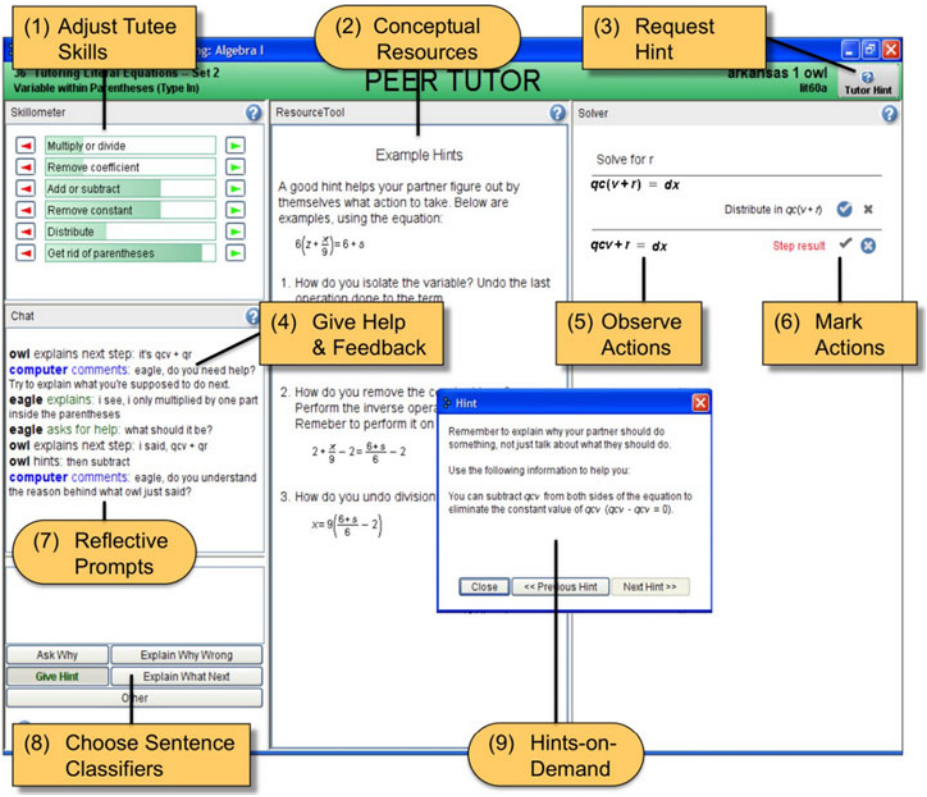

Fig. 3 Peer tutor's interface. Square labels represent possible peer tutor actions in the interface. Round labels represent the support peer tutors receives from the adaptive system

hint, they could click “give hint” and then type “subtract x”. Their utterance would appear as: “tutor hints: subtract x” to both students in the chat window. Tutees were also asked to self-classify each utterance as one of three categories: a “ask for help”, “explain yourself”, or “other”.

We attempted to trigger peer tutor reflective processes by providing tutors with adaptive domain assistance that supported them in identifying and reflecting on tutee errors. We intended that this assistance have the additional benefit of ensuring that the tutee received more correct help than they otherwise would have. We implemented cognitive help for the peer tutor 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 (e.g., “Your partner is actually wrong. Here is a hint to help you explain their mistake.”), and the domain help the tutees would have received had they been solving the problem individually (e.g., “Subtract *x* to move to the other side.”). Further, 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. Students were not allowed to move to the next problem until the current problem was successfully completed.

Including adaptive help-giving assistance in APTA

In the current system we augmented this cognitive assistance with three types of help-giving assistance, designed based on the principles identified in Phase 1. The first type of assistance, *hints on demand*, is used for instances when the peer tutor (for convenience, we will call her Sara) does not know how to help the tutee (we will call him James). There may be moments where James has asked for help, and Sara does not know what the next step to the problem is or how best to explain it. In this case, Sara would click on a hint button, found in the top right corner of the interface (#3 in Fig. 3), and receive a multi-level hint on both how to solve the problem and how to help her partner. The hint opens with a collaborative component (“Remember to explain why your partner should do something, not just what they should do”), and then contains the cognitive component that the tutee would have originally received had they been using the CTA individually (“You can subtract *qcv* from both sides of the equation to eliminate the constant value of *qcv* [*qcv –qcv = 0*]“; see #9, Fig. 3). If Sara still doesn’t understand what to do and clicks next hint, both the collaborative and the cognitive component become more specific, until the cognitive component ultimately reveals the answer to Sara. The collaborative component uses several strategies to encourage students to give more conceptual help, and is adaptively chosen based on the current problem-solving context (e.g., it varies depending on whether the tutee has most recently taken a correct step or an incorrect step). Sara is intended to integrate the cognitive assistance for how her tutee, James, should proceed in the problem with the collaborative assistance on what kind of help she should give. In this case, Sara might use the information she received to tell James “Eliminate the constant value of *qcv*”. This hint does not reveal the answer to the tutee, but includes relevant and correct domain content.

There may be cases where even after examining the adaptive hints, Sara is still unsure how to use the hints to give the tutee appropriate feedback (e.g., how to give help that refers to information James already knows). We designed the *adaptive resources* to further assist the peer tutor in constructing good help. When Sara clicks the "give hint" sentence classifier to prepare to compose a hint to her partner (#8 in Fig. 3), she is presented with a resource (#2 in Fig. 3), with content tailored to the current problem type, which provides examples of what a good hint would be within the context of this problem type. We had four separate sets of resources mapping to each type of sentence classifier (one for "ask why", one for "explain why not", one for "give hint", and one for "explain next step"). As the resource presents several sample hints for the whole problem, Sara has to actively process the resource in order to determine which kind of hint might apply to the information she has to convey. We expected that Sara would use the adaptive hints and resources together to construct help.

Once Sara has given help to her partner, she might receive a *reflective prompt* in the chat window that appears simultaneously to both students and targets peer tutor help-giving skills that need improvement (#7 in Fig. 3). For example, if Sara is a novice tutor she may give a novice hint like "then subtract" rather than a conceptual hint like "to get rid of *qcv,* you need to perform the inverse operation on that side of the equation." In that case, the computer uses its assessment of Sara's help-giving skill to say in the chat window (visible to both Sara and James), "James, do you understand the reason behind what Sara just said?" This utterance is designed to get both James and Sara reflecting on the domain concepts behind the next step, and to remind Sara that she should be giving help that explains why in addition to what. Prompts could be addressed to the peer tutor (e.g., "Tutor, can you explain your partner's mistake?") or the tutee (e.g., "Tutee, do you know what mistake you made?"), and were adaptively selected based on the computer assessment of help-giving skills (see below). They contained both praise and hedges, such that the computer's voice never publicly threatened the peer tutor's voice. Students also received encouragement when they displayed a particular help-giving skill (e.g., "Good work! Explaining what your partner did wrong can help them not make the same mistake on future problems"). In addition to receiving prompts related to the help given, there were prompts encouraging students to use sentence classifiers more effectively (e.g., "The buttons underneath the chat [e.g., "Give Hint"] can help you let your partner know what you're doing"). Only one reflective prompt was given at a time, and parameters were tuned so that students received an average of one prompt for every three peer tutor actions. There were several different prompts for any given situation, so students rarely received the same prompt twice.

In order to decide when to give students reflective prompts, we built a model for good peer tutoring which assessed whether students displayed four help-giving skills: help in response to tutee errors and requests, help that targets tutee misconceptions, help that is conceptual and elaborated, and the use of sentence classifiers to give help. Our main focus was on supporting peer tutors in giving conceptual elaborated help, and we discussed how the use of sentence classifiers might facilitate that by encouraging peer tutors to reflect more on the help they give. Similarly, by encouraging peer tutors to target tutee misconceptions, we encouraged them to reflect and elaborate more on the concepts involved in solving the problem in general. Finally, by encouraging peer tutors to give help when tutees need it, we actively discouraged them from giving instrumental help after tutees take correct steps, a common approach that students take. While these skills should also benefit tutee learning from peer tutoring, our primary focus for the time being was on the elaborative processes triggered by peer tutor help-giving, and the learning that might ensue. To assess peer tutor performance, the model used a combination of several inputs. First, it used CTA domain models to see if tutees had recently made an error (and thus if they needed help). Next, it used student interface actions, including tutor self-classifications of chat actions as prompts, error feedback, hints, or explanations, to determine what the students' intentions were when giving help. Finally, it used *TagHelper* (Rosé et al. 2008), a toolkit for automated text classification that has been successfully deployed in educational contexts, to build a machine classifier trained on previous study data (Cohen's kappa = .82 for the previous dataset). The classifier could automatically determine whether students were giving help, and whether the help was conceptual. Based on a combination of these three sources of information, we used a simple computational model composed of 15 rules to assess each peer tutor action taken. We used Bayesian knowledge tracing (Corbett and Anderson 1995) to update a running assessment of peer tutor mastery of conceptual help, targeted help, timely help, and appropriate use of sentence classifiers. If, after any given peer tutor action, a given tutor skill fell within a predefined threshold for that skill, students were given a reflective prompt targeting that skill. We defined thresholds both for positive and negative feedback.

Discussion

We expected APTA to have several positive cognitive and motivational effects. We designed adaptive support to give students relevant knowledge on how they can improve the conceptual content of their help when they can apply it. Additionally, as students using APTA received a lot of assistance that directly encourages students to use classifiers appropriately, we expected that the adaptive assistance would also have a positive effect on how often and how accurately students used classifiers. We were further interested in exploring how the way we designed the adaptive support motivated students to interact with and apply the support, based on the principles derived in Phase 1. We hypothesized that we had engaged the *Relevance Design Principle* in the hints on demand, where we gave a collaboration hint in conjunction with a context-relevant cognitive hint. By seeing immediately how the collaboration hint applies to the cognitive hint, students may perceive the collaboration hint as more relevant. This principle is also applied in the adaptive resources, which are directly linked to student choice of sentence classifier, ideally motivating students to interpret and apply the resources. The *Accountability Design Principle* was engaged in the adaptivity of the conceptual resources and in the reflective prompts. The reflective prompts were presented publicly, putting more of an onus on the peer tutor to follow the prompt. This feeling of accountability was likely augmented by the way the resources changed as students select different classifiers, suggesting to students that they should be putting thought into the help they give. We employed the *Efficacy Design Principle* by formulating prompts in ways that provide positive feedback and add to what the peer tutor is trying to do, rather than contradict it.

Phase 3: In vivo experiment

We deployed APTA in a classroom experiment to examine the influence of the adaptive support on peer tutor help-giving behaviors. In order to determine whether it was indeed the way the support was designed that was producing a change in student behavior, we compared it to fixed support that provided the same collaborative instruction, but did not include adaptive elements. In this section, we describe a quantitative analysis of the effects of the adaptive support as compared to fixed support on interaction and learning. In the following section (Phase 4), we discuss a qualitative analysis exploring to what extent our designs for accountability, efficacy, and relevance had the desired impact.

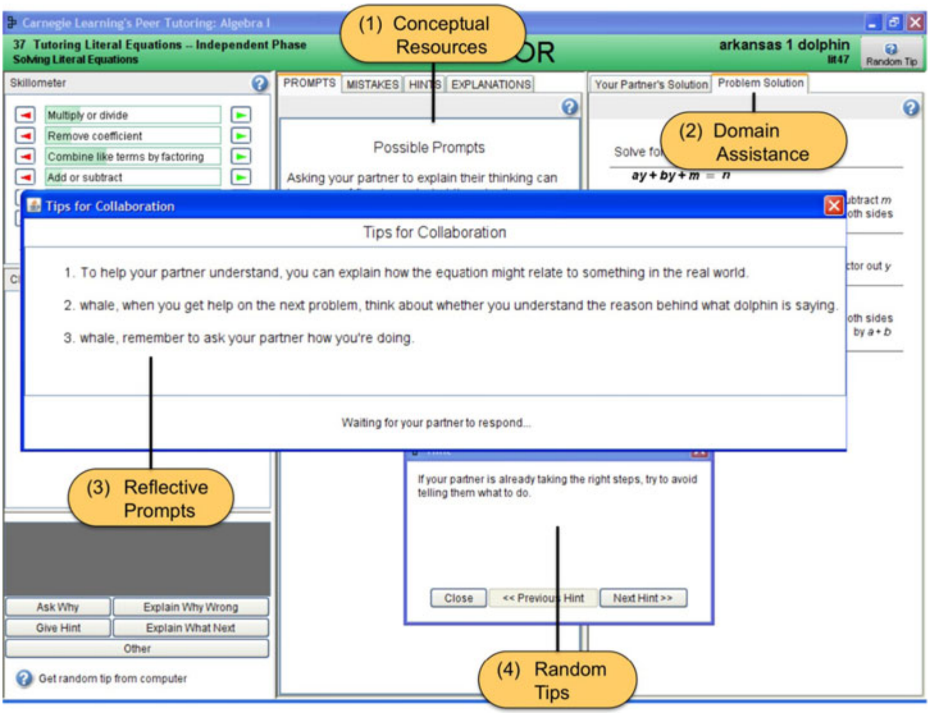

Fig. 4 Peer tutor's interface in fixed support condition. Conceptual resources are not connected to sentence classifiers, domain assistance is in the form of fixed problem solutions, reflective prompts are randomly delivered between problems, and the students can request randomly selected collaboration tips

Study conditions

The *adaptive support condition* included the adaptive resources, and reflective prompts described in the previous section. Furthermore, it included the traditional *CTA* hints on demand, and the cognitive hints and feedback from the previous version of *APTA*. The *fixed support condition* contained the same support content as the adaptive system, but the content was not presented adaptively (see Fig. 4). To create a fixed parallel to the *adaptive cognitive support*, where peer tutors were given domain hints and feedback, we provided students with annotated solutions to the current problem (#2 in Fig. 4), a technique that had been used as part of other successful peer tutoring scripts (e.g., Fantuzzo et al. 1989). With this fixed assistance, peer tutors could consult the problem solutions at any time, but would not receive feedback on whether their help was correct or whether the current problem was completed. To parallel the *hints on demand,* we gave students access to a "Random Tip" button that yielded multi-level randomly selected tips (#4 in Fig. 4). While the overall content of tips was the same as the hints on demand, the tips were randomly selected rather than chosen adaptively. The random tips did not contain any adaptive cognitive content. For *adaptive resources,* we gave students access to the same resources as they had in the adaptive condition, but the resources did not change based on the sentence classifiers students selected—instead, students had to select which resource they wanted to view without additional guidance (#1 in Fig. 4). Finally, instead of receiving *reflective prompts* in the chat window, we gave students reflective collaborative tips between each problem, with parallel content to the reflective prompts present in the adaptive condition (#3 in Fig. 4). Each student was presented with 5 randomly chosen reflective statements after each problem was complete such as “Good work! Remember, hinting or explaining the reason behind a step can help your partner learn how to do the step correctly.” We chose that form of support also because it is common for students using a collaborative script to receive reflective prompts at fixed intervals. This approach was a reasonable way to provide students with similar content to the adaptive condition. A summary of support is shown in Table 1.

Method

Participants

Participants were 104 high-school students (54 male, 50 female) from two high schools, currently enrolled in Algebra 1, Algebra 2, or Pre-Calculus. Both high schools used the individual version of the *CTA* as part of regular classroom practice so students were used to working with the tutors. The literal equation solving unit that we used was a review unit for the students, and one that they had already covered in Algebra 1. Nevertheless, the concepts in the unit were difficult for the students to understand, and teachers were in favor of reviewing the unit. Students from each class were randomly assigned to one of the two conditions, and to either the initial role of tutor or tutee (later they switched roles). For the purposes of this analysis, we are interested in those students who interacted with the system as a tutor, and thus excluded 27 students who only took on the role of tutees; that is, they were absent on one or both supported tutoring days and were tutees on the days they were present. We further excluded one student who was partnered with a teacher when tutoring, and two students who played the role of tutor in both collaboration periods. A total of 74 students were included in the analysis.

Procedure

The study took place over the course of a month, spread across six 45-minute classroom periods. During the first period, students took a 15-minute pretest measuring

Table 1 Assistance differences in adaptive and fixed systems

Assistance Type	Adaptive System	Fixed System
Cognitive Feedback	Must finish a problem before moving on. Receive feedback on marking actions.	Move to next problem when students believe current problem is complete. Seek out feedback on marking actions by accessing problem solutions.
Peer-Mediated Hints	Receive integrated cognitive & interaction hint	Seek out a cognitive hint by accessing problem solutions. Receive list of interaction tips at the end of each problem.
Conceptual Resources	Get resource linked to selected sentence classifiers.	Select resources and use classifiers independently.
Reflective Prompts	Receive reflective prompts based on dialog.	Receive reflective prompts while waiting for next problem to load.

domain learning. Then, in the second period, students spent 45 min in a preparation phase, solving problems individually using the CTA. Students worked on one of two problem sets, focusing on either factoring in literal equation solving or distributing in literal equation solving. The third and fourth periods were collaboration periods, where students were given partners, and tutored them on the problems they had solved in the second period, with either adaptive or fixed support. Students were given different partners for each of the two collaboration periods. They were paired with students who were in the same condition, but who had solved a different problem set during the preparation phase. Within these constraints, we assigned pairs randomly, with the exception of not pairing students teachers explicitly told us would not get along. Within a pair, students were randomly assigned to the tutor or tutee role during the first collaboration period, and then they took on the opposite role during the second collaboration period. In the fifth period, students collaborated with new partners without any adaptive support (as an assessment of their collaborative abilities) and in the sixth period, between 2 and 3 weeks after the completion of the study, students took a posttest to assess their domain learning.

Measures

To assess students' individual learning we used counterbalanced pretests and posttests, each containing 10 conceptual items, 5 procedural items, and 2 items that demanded a verbal explanation. Some of the conceptual items had multiple parts. The tests were developed by the experimenter, but adapted in part from Booth and Koedinger's measures of conceptual knowledge in Algebra (2008). Tests were approved by the classroom teacher, and were administered on paper. We scored answers on these tests by marking whether students were correct or incorrect on each item part, computing the scores for each item out of 1, and then summing the item scores to get a total score.

In order to analyze student collaborative process, we logged all semantic actions students took within the system, including tutee problem-solving actions, sentence classifiers selected by both students, and chat actions made by both students. Along with the student actions, we logged computer tutor responses, which includes both the system's evaluation of the action and the assistance students received. Using this data, we computed the number of problems viewed by each student, and the number of problems correctly solved (in the fixed condition, students could move to the next problem without having correctly solved the previous one). We calculated the number of errors viewed by students when they took on the peer tutoring role, and the number of times peer tutors used each type of sentence classifier. Finally, we computed peer tutor exposure to the assistance in our system, including the number of times they received reflective prompts and the number of times they requested a cognitive hint.

We segmented the dialog by chat messages (creating a new segment every time students hit enter), and two raters coded the chat data on several dimensions. We computed inter-rater reliability on 20% of the data, and the remainder of the data was coded by one rater and checked by a second. All disagreements were resolved through discussion. First, each help segment was coded for whether it constituted previous-step help, that is, help relating to an action tutees had already taken (e.g., "no need to factor because there is only one *g*"; Cohen's kappa = 0.83), or whether it was next-step help, that is, help relating to a future action in the problem (e.g., "how would you get rid of $2h$?"; Cohen's kappa = 0.83). Finally, each help segment was coded for whether it contained a concept (e.g., "add *ax*" is purely instrumental help, while "add *ax* to cancel out the –*ax*" is conceptual). Cohen's kappa for conceptual help was 0.72.

Quantitative results

We used quantitative interaction and learning data to determine if the peer tutor's help quality increased because of the assistance they received, and if an increase in help quality translated into a learning improvement.

Overall interaction context

First, to get a sense of the context of student interaction, we examined whether there were systematic high-level differences between the two conditions in the way students solved problems and gave help. We used a MANOVA with condition as the independent variable to evaluate the differences between conditions for the following variables: problems viewed, problems completed correctly, tutee errors viewed by tutors, and help given by tutors. The analysis revealed significant differences between conditions (Pillai's Trace = 0.30, F [1, 72]= 7.68, p=0.001). Table 2 displays the results of one-way ANOVAs for each dependent variable. The students in the fixed condition saw significantly more problems than students in the adaptive condition (row 1). Students in the fixed condition could skip past problems that gave them trouble (and occasionally did not realize they had made a mistake), while students in the adaptive condition had to overcome every impasse they reached. However, both conditions completed similar numbers of problems correctly (row 2), and the total number of tutee errors viewed by peer tutors was not significantly different across conditions (row 3). Finally, the amount of help given by peer tutors was not significantly different across conditions (row 4). The ratio between errors viewed by the peer tutor and help given was roughly 4:3 in the adaptive condition and 1:1 in the fixed condition. In the following, we present count data of particular aspects of student interaction, and use negative binomial regression to test the relationship between variables. Unless otherwise noted, we will perform statistical tests on the raw data counts, but to better illustrate what occurred, we may also present ratios between the count data and context variables like errors viewed or total amounts of help.

Amount of conceptual help

Our main goal in the design of the adaptive assistance was to improve the quality of help given in the adaptive condition. This goal was operationalized as improving the amount of conceptual help given, since conceptual help is an indicator of the elaborative processes in

Table 2 Differences in problem-related actions across conditions

Context variables	Adaptive		Fixed		ANOVA results	
	M	SD	M	SD	$F(1, 74)$	p
Problems seen	7.90	4.51	10.43	5.76	4.483	0.038
Problems completed	7.26	4.42	7.60	4.37	0.113	0.738
Errors viewed	15.71	8.56	12.63	8.41	2.444	0.122
Help given	11.92	6.23	12.17	8.87	0.019	0.890

Table 3 Differences in help quality between conditions, as measured by the amount of conceptual help given and the way peer tutors used classifiers

Interaction variables	Adaptive		Fixed		Mann-Whitney results	
	M	SD	M	SD	U	p
Conceptual help (n=74)	2.67	2.83	1.34	2.14	468.5	0.015
Classifiers used (n=74)	7.95	6.77	4.28	5.78	371.5	0.001
% help given with classifiers (n=71)	56.8%	33.7%	31.0%	34.4%	348.00	0.001

peer tutoring, and a predictor of learning gains for both students. The effects of condition on conceptual help were significant (see Table 3, row 1). In total, roughly 20% of the help was conceptual in the adaptive condition, nearly double the percentage of help that was conceptual in the fixed condition (10%).

Frequency and accuracy of classifier use

In addition to improving the quality of student help-giving, we intended that the adaptive help would improve student use of interface features, and in particular, encourage students to use the sentence classifiers while chatting. As described in the Introduction section, sentence classifier use is theoretically related to help quality, and thus should be related to the amount of conceptual help that students give. Further, the more appropriately students use classifiers, the better intelligent systems are determining the content of student chat. Thus, one hypothesis we had was that students would use help-related classifiers (i.e., not the neutral "other" classifier) more frequently in the adaptive than in the fixed condition, regardless of the content of their utterances. This hypothesis was supported by the data (see Table 3, row 2). Students used roughly 2 classifiers for every 3 errors in the adaptive condition, compared to 1 classifier for every 3 errors in the fixed condition. However, while this measure reflected how often students used classifiers, it did not reflect the student's purpose in using the classifiers. We also predicted that when peer tutors gave help to tutees, they would be more likely to label their utterance with one of the help-related classifiers than the "other" classifier. The percentage of help given using help-related classifiers was significantly greater in the adaptive condition than in the fixed condition (see Table 3 row 3), suggesting that students used classifiers appropriately more often in the adaptive condition. The percentage of non-help chats given using help-related classifiers were not significantly different between conditions, suggesting that it was not increased classifier use overall that was driving the effect.

We further explored the relationship between condition, sentence classifiers used, and conceptual help given. The number of classifiers used and conceptual help given were correlated ($r[72]=0.442$, $p<0.01$), but it was not clear whether condition had separate effects on classifiers used and conceptual help given, or whether the number of classifiers used influenced the amount of conceptual help given (as suggested by prior research on sentence classifiers). To explore these separate possibilities, we conducted a regression analysis to predict the amount of conceptual help given controlling for the number of classifiers used. We used student condition, the number of sentence classifiers used, and the amount of help given overall as predictor variables. The model was indeed statistically significant ($\chi^2(3, N=74)=33.287$, $p<0.001$). Condition was a significant predictor of conceptual help given ($\beta=0.687$, $\chi^2(1, N=74)=5.84$, $p=0.016$), as was the amount of help given (β=0.087, χ^2(1, N=74)=22.97, p<0.001). Classifiers used were marginally predictive (β=0.042, χ^2(1, N=74)=3.78, p=0.052). Based on these results, when help given and classifiers used are held constant, the adaptive condition is responsible for about 1.98 more instances of conceptual help than the fixed condition. This analysis suggests that while both help given and classifiers used were predictive of conceptual help given, condition had an independent effect.

Learning outcomes

Finally, we looked at whether learning outcomes varied between the two conditions. The adaptive condition had a mean pretest score of 33.53% (SD=25.11%) and posttest score of 40.55% (SD=21.50%). The fixed condition had a mean pretest score of 39.13% (SD=23.92%) and posttest score of 47.10% (SD=26.28%). We conducted a two-way repeated-measures ANOVA with condition as a between-subjects variable and test-time as a within-subjects variable. We used only students who had participated in the pretest, posttest, and an intervention phase as a peer tutor. All students learned (F [1, 49]=11.97, p=0.001), but there were no significant learning differences between conditions (F [1, 49]=0.048, p=0.828).

Phase 4: Case studies

The adaptive support improved two aspects of peer tutor help given compared to the fixed condition: conceptual content and use of sentence classifiers. We next investigated to what extent the positive influence of the adaptive support on help-giving was related to the hypothesized desired effects on student motivational factors, following the design principles identified in Phase 1. We present one case representative of the positive effects of accountability on student help-giving, and one case representative of the negative effects of a lack of perceived relevance. We use both cases to discuss the influence of efficacy on student help-giving. While there are likely many contextual factors contributing to the influence of support on student help-giving, we limit our discussion here to the three identified principles, in order to better follow up on the Phase 1 results.

A case of accountability & elaborative processing

With this case study of Dyad 1, we illustrate how feelings of accountability to be good tutors engendered by the adaptive support encouraged dyads to engage in elaborative processing. In this dyad, the peer tutor scored 55% on the posttest, and the tutee scored 20%. The interaction occurred on the second tutoring day, and concerned the problem $kj - mk = fr$, solve for k. It was the second problem the dyad had seen that day, but the first with this form. Over the course of the interaction, the different assistance types increased the peer tutor's accountability to knowledge and to reasoning – that is, her effort to give the correct answer and to give a conceptual explanation for her answer. The interaction begins with the tutee asking for help (see Table 4, row 1). When the peer tutor clicks on the sentence classifier "explain next step" to compose her response, the peer tutor receives a resource on how to construct good explanations. On first glance, the resource appears to have little effect, as only 10 seconds pass between the time the resource is presented in the interface and the time the peer tutor's response is submitted, and the peer tutor gives

Table 4 Case study demonstrating the positive effects of adaptive support on student interaction. Support may trigger student feelings of accountability

Solve for *k*: $kj - mk = fr$			
1	Tutee:	[both]	What should I do first?
2	Tutor:	[self]	*chooses "explain next step" classifier*
3	Agent:	[tutor]	*gives resource on explanations*
4	Tutor:	[both]	Add *mk* to both sides.
5	Agent:	[both]	Tutee, do you understand the reason behind what the tutor just said?
6	Tutee:	[both]	*adds mk to both sides of the problem*
Solve for *k*: $kj = fr + mk$			
7	Tutee:	[both]	Does it matter that there's a k on the right side?
8	Tutor:	[both]	*marks the "add mk" step correct*
9	Agent:	[tutor]	*highlights step*
10		[tutor]	This step is wrong. Give your partner some advice on what to do next
11	Tutor:	[tutor]	*chooses "comments" classifier*
12		[both]	Wait!!! I completely messed up... the computer wants you to subtract *kj* from both sides, because of the other *k* in the problem. sorry = (
13	Tutee:	[*both*]	haha, it's alright, these problems are so simple but confusing.

instrumental help ("add *mk* to both sides"; row 4). However, the simple presentation of this resource begins to establish the expectation that peer tutors are expected to put thought into the help that they give. A second type of assistance is presented immediately after the peer tutor has delivered her instrumental help: the computer says in the chat window, where both collaborators can see it ("Tutee, did you understand the reason behind what the tutor just said?"; row 5). Not only is the computer prompting the tutee to reflect, but it is also publicly reminding the peer tutor that help should include an explanation in addition to an instruction, further increasing the peer tutor's accountability for giving elaborated help. In fact, the tutee responds to this prompt with evidence of deep processing (row 7): "Does it matter that there's a *k* on the right side?" The tutee is reflecting on features of the problem that are relevant for attaining the problem solution. After the tutee has in fact added *mk*, and the peer tutor has marked the step wrong, the computer further enforces the peer tutor's accountability to provide help by saying privately to the peer tutor: "This step is wrong. Give your partner some advice on what to do next." At this point, the peer tutor's response represents a breakthrough in the peer tutor's helping behaviors. The peer tutor responds with a conceptual statement, saying "the computer wants you to subtract *kj* from both sides, because of the other *k* in the problem" (row 12). This statement explains what the tutee should do, explains why, and alludes to the concept that all *k*s in this problem have to be on the same side, suggesting that the peer tutor is reflecting on the next step and elaborating on her knowledge. It is the first conceptual statement made by this particular peer tutor. This insight on the part of the tutor, and articulation of the insight to the tutee, had benefits for both parties. The error that Dyad 1 made during this problem required them to master the concept that to solve for a given variable all instances of the variable need to be moved to the same side of the equation. Both the tutor and the tutee in the dyad got a similar problem correct on the individual posttest, suggesting that as a result of this interaction, they had mastered the discussed concept.

Table 5 Case study demonstrating the problem with perceived relevance of adaptive support. While the peer tutor perceived domain support as relevant, he did not use the interaction support

Solve for *q*: $(6\ t - wr)/vt = (qt + qv)/vt$			
1	Tutee:	[both]	It won't let me get rid of the v and t. Help me.
2	Tutor:	[agent]	*requests hint*
3	Agent:	[tutor]	A good hint explains why in addition to what. What can you do to both sides to get the q by itself? In $(qv + qt)/vt$, $qv + qt$ is divided by vt. How do you undo division? Multiply both sides by vt.
4	Tutor:	[self]	*chooses "give hint" classifier*
5	Agent:	[tutor]	*gives resource on hints*
6	Tutor:	[both]	Multiply both sides by *vt*
7	Agent:	[both]	Tutee, can you talk about why you took that last step?
8	Tutee:	[both]	What last step? The simplifying fractions?
9	Tutor:	[both]	*marks the simplifying fractions step right*
10	Agent:	[tutor]	This step is wrong. Get your partner to think about what to do next.
11	Tutor:	[agent]	*requests hint*
12	Agent:	[tutor]	Remember to explain what your partner did wrong, in addition to what to do next. What can you do to both sides to get the *q* by itself? In $(qv + qt)/vt$, $qv + qt$ is divided by vt. How do you undo division? Multiply both sides by vt.
13	Tutor:	[self]	*chooses "give hint" classifier*
14	Agent:	[tutor]	*gives resource on hints*
15	Tutor:	[both]	Delete the last 3 steps and multiply both sides by *vt*

A case of support relevance & shallow processing

While peer tutors appeared to find adaptive help on how to solve the problem extremely relevant, they did not have a similar response to adaptive assistance on how to give good help, potentially leading them to process the problem shallowly. The case of Dyad 2 in Table 5, who engaged in suboptimal interaction, is from the first tutoring day (the third study period); the dyad was solving the problem $6t - qt = wr + qv$. This was their ninth problem of the day, but the first problem they had encountered where they had to move two variable instances to the same side. The peer tutor had scored 23% on the pretest, and the tutee had scored 38%. The following dialogue begins when the tutee had reached the equation $6\ t - wr = qt + qv$, but then incorrectly divided both sides by vt instead of $v + t$. The tutee triggers the exchange using a question that shows the tutee is reflecting on the situation ("It won't let me get rid of the *v* and *t*. Help me"; row 1). The peer tutor asks for a hint, but then only transfers the instrumental component of the hint to the tutee ("Multiply both sides by *vt*"; row 6), suggesting that while the peer tutor feels that the domain help is relevant, he doesn't perceive the conceptual scaffolding as relevant. As in the previous scenario, the computer prompts the tutee for further explanation ("eagle, can you talk about why you took that last step?"; row 7), but this only serves to confuse the students further ("what last step?"; row 8), suggesting that the vague wording of the prompts may be a liability in this case. After getting more content-related feedback, and another hint, the peer tutor relays the hint to the tutee again (row 15). After this exchange, the tutee realizes his error and proceeds to solve the problem, without interacting further with the peer tutor. This lack of communication has effects on the posttest results: For Dyad 2 to solve this problem correctly, they needed to master the concept that to isolate the x in an expression like $x(a + b)$ you need to divide by $(a + b)$. Neither member of the dyad got the related conceptual question right on the posttest. Not surprisingly, the peer tutor in this interaction came out of the session unsure of how to use the computer help, saying in the following period when he was the tutee: "yeahh the tutor is confusing cuz it gives youu all this stuff to write about but I had no clue what to write when i was the tutor." In summary: When this peer tutor gave help, he ignored the collaborative components of the hint he received and focused on the cognitive component, which contradicts what we had intended with our design.

Peer tutor self-efficacy: Transfer of control

We had also attempted to design feedback to maintain the peer tutors' sense of tutoring self-efficacy. To a certain extent, the design appeared successful, and in some cases, the support we gave helped peer tutors to take control of the situation. For example, after one tutee added ax to both sides in the problem $ax - y=8$, and the peer tutor marked it right, the peer tutor received the feedback: "This step is not right. Tell your partner what mistake they made. Here is a hint to help you tutor your partner. Since $a*x$ is positive, you should subtract to remove it from the left side. Erase your last step and subtract $a*x$ from both sides." In consequence, the peer tutor changed their response, marking the step wrong, and then smoothly gave the conceptual hint "It's a positive *ax* you wouldn't add u would subtract." This peer tutor was adept at using the cognitive tutor hints to give their partner guidance, and while the peer tutor didn't acknowledge his error, he did give error feedback to the tutee. However, sometimes students would attribute hints to the computer in order to indicate their uncertainty and to convey to their peer tutee a sense that they (peer tutor and tutee) are in the same boat. A good example of this phenomenon is in the first case study, where the peer tutor both attributes the hint to the computer, and apologizes for the confusion ("wait!!! I completely messed up... the computer wants you to subtract *kj* from both sides, because of the other *k* in the problem. sorry = ("). Interestingly, the peer tutor gives a much more elaborated hint than the one she had received from the computer, but still attributes the hint to the computer, probably to indicate her own lack of confidence in the solution. The same students from Dyad 1 verbally expressed similar sentiments at several points, bonding over their own inexpertise: The peer tutor said, "wow... this is so confusing!" The peer tutee replied, "I'm glad I'm not the only one who's confused! hahaha". Those two students went on to be successful at solving the problem. Against our designs, it appeared that the peer tutor indicating uncertainty and attributing help to the computer was beneficial for the tutor-tutee relationship, suggesting that in some cases, constant maintenance of peer tutoring efficacy was not as necessary as the peer tutor being able to transfer control to the computer, shifting between expert and novice.

General discussion

In this paper, we described a four-phase design process for developing adaptive assistance for help-giving in a peer tutoring context. First, we used Speed Dating, a human-computer interaction design method, to generate three principles for designing adaptive support for collaborating students. On the basis of the three principles we developed three types of help-giving assistance for our peer tutoring system: hints on demand, conceptual resources, and reflective prompts. We evaluated the resulting system in an in vivo experiment, and found that compared to a fixed support condition, the adaptive assistance improved the conceptual content of student help and their use of sentence classifiers. Case analyses of process data of two dyads from the adaptive condition suggested that while we successfully designed to increase student accountability, we were not as successful at increasing the perceived relevance of the adaptive collaboration support. In this section, we discuss the theoretical conclusions and design implications of our empirical results, and the promise of iterated in vivo experimentation.

Theoretical conclusions

Our results add to the small but growing body of evidence that adaptive support can improve the quality of student collaboration. Previous research in the effects of adaptive support compared to a fixed control has found that adaptive support can increase student learning (Kumar et al. 2007), but little is known about how adaptive support affects collaborative process. Our research provides direct evidence of the effects of adaptive support on a specific facet of student interaction: tutor help-giving. We found that the adaptive support led students to increase the conceptual help content of their utterances compared to fixed support. Help-giving is one of the positive aspects of collaboration specified by Johnson and Johnson (1990), and conceptual content is widely recognized as an important component of help-giving (Fuchs et al. 1997; Webb and Mastergeorge 2003). Thus, our intervention (and in particular, the use of adaptive reflective prompts) could potentially be applied to other collaborative scenarios that would benefit from an improvement in the conceptual quality of student help-giving in text-based chat (such as the Virtual Math Teams system; Stahl 2009). Our system also used simple adaptive prompts to improve the way peer tutors used sentence classifiers for their help-giving, in that they chose to use help-related sentence classifiers more often and more accurately. Applying a similar method to other systems that incorporate sentence classifiers and starters may make those other interventions more effective. For example, introducing reflective prompts into the *GroupLeader* system may improve the difficulties the system developers have encountered in getting students to use sentence starters accurately (Israel and Aiken 2007). We did not find effects of adaptive support on student learning, compared to fixed support, but our study was a short-term study, and attrition between the intervention and the posttest was rather large. In theory, well-designed adaptive support will, in the end, have a positive effect on student learning.

Our design principles, informed by our qualitative results, contributed to the research into how students are motivated by peer tutoring. While most studies have looked broadly at how reward structures increase student accountability (e.g., Fuchs et al. 1997), they have not examined how this mechanism might be working *during* the interaction. Our design work in Phase 1 and the qualitative analyses in Phase 4 support the conclusions of previous experimental manipulations by demonstrating that students feel accountable to be good peer tutors to their partners, and that this accountability increases when relevant and public support is given to tutors (i.e., when peer tutor responsibility is primed). With the increase in accountability, students put more effort into constructing help and applying the assistance they received to their help, potentially engaging in more cognitive elaborative processes. This result suggests that it may not be the *adaptivity* of the support that is creating these results, but the *perceived adaptivity*. In other words, a different fixed control that takes the form of random prompts in the chat window may have the same effect as adaptive support, if students perceive the prompts as adaptive and are thus motivated to feel more accountable for the help they construct.

Given this analysis, and the effects of the support on student help behavior but not on learning outcomes, it is still an open question whether the effort and expense required to develop adaptive systems for collaboration is worth the result. Our classroom study had several limitations, including that the sample size was relatively small, and that the experimental manipulation included different types of support, which may have made the system unnecessarily complicated for students to use. Another study is necessary to tease apart to what extent adaptivity has cognitive benefits (i.e., students get the support at the correct time and thus benefit more), and to what extent it has motivational benefits (i.e., students feel accountable for incorporating support because they believe it is adaptive). Ideally, this study would have a larger sample size, take place in a more controlled environment, and vary only one type of adaptive support. If the results of this future study suggest that adaptive support only has motivational benefits, and that the adaptiveness itself does not lead to learning, then it is likely that research on supporting collaboration should focus elsewhere. On the other hand, if the results of this future study suggest that adaptivity is important and adaptive support has learning benefits, research on achieving adaptivity of support should indeed be encouraged, despite the expense.

Design implications

Our results also have direct implications toward defining a design space for ACLS that can inspire future research. As described in the introduction, most ACLS support is very similar in the way it provides feedback to students. This similarity can be conceptualized on two dimensions: the feedback is usually *directly* delivered to the ineffective collaborator, and it is *explicit* with respect to telling the ineffective collaborator what they did wrong and how to fix it. The assistance used as part of needs identification in Phase 1 and incorporated in our system in Phase 2 covers a much broader design space. The hints on demand we provide has a direct and explicit component, but also a peer-mediated component: we expect the peer tutor to benefit from receiving domain help and communicating it to the tutee. The conceptual resources are directly presented to the peer tutor, but they are implicit in nature: The peer tutor is expected to read the content and determine how to use it to generate good help in that particular situation. The reflective prompts we included in the system are situated somewhere in the middle of the design space, they are presented to both students, and worded in a general way.

While it is difficult to directly compare different assistance types in our experimental design, we can draw some links to research questions about how varying explicitness and directness might impact the adoption of support. First, how does directness influence accountability? In our study, students appeared to feel more accountable to use support meant for their partner (i.e., the cognitive hints) or support that was publically delivered (i.e., the reflective prompts) than support that was delivered directly and only to the peer tutor (i.e., the collaborative portion of the hints, and the adaptive resources). Future studies could tease out the effects of directness from the effects of support type. A second research question might be: What effects does explicitness have on relevance and efficacy? While students appeared to find the most explicit support to be the most relevant, particularly in Phase 1, peer tutors also resented any support that undermined their ability to tutor. In general, by developing a sense of how different features of adaptive collaboration support affect how students react, we can design effective adaptive support tailored to particular student populations and contexts.

Iterated in vivo experimentation

To conclude, we would argue that the iterated *in vivo* experimentation design process we have described in this paper represents a fruitful combination of design research and controlled experimentation. It is true that this approach has certain drawbacks; we lack the full ecological context and deep evaluation that design research brings, and we lack the full control provided by psychological experimentation. However, this approach has unique value in that it effectively finds a balance between tradeoffs commonly found in learning sciences research. Our approach balanced experimental control and ecological validity by allowing us to draw conclusions about how adaptive support affects student behavior while maintaining a holistic perspective. For example, our choice to evaluate multiple types of adaptive assistance simultaneously represented a loss in experimental control, as we varied multiple dimensions of support. However, it allowed us to examine the effects of each type of assistance through an analysis of the process data, and thus draw richer conclusions about how support might affect student motivation. Similarly, our use of mixed methods in combining quantitative data with qualitative analyses allowed us to link accountability to the quality of student help-giving in a way that would be difficult had we not combined the approaches. Without the qualitative data it would have been difficult to determine why student use of conceptual help improved; but without the quantitative data, it would have been difficult to determine how differences between isolated cases mapped to systematic differences between conditions. Finally, one of the contributions of our particular process is that, through our use of human-computer interaction design methods such as Speed Dating, we treated the students as “users” instead of just “learners”. We designed for use by analyzing how students perceived and reacted to the support, instead of solely examining which support was likely to lead to the most learning. This approach served as a good precursor to designing a form of assistance students do not typically receive, and letting us know what to expect in terms of how students interact with that kind of assistance. By designing a system that responded to student motivational needs, we hoped to ultimately have a more positive effect on their learning as well. In summary, as a result of our approach of iterated *in vivo* experimentation, we have made theoretical contributions to the literature on adaptive support for collaborative learning, and defined a space for future experimental and design research.

Acknowledgments This project is supported by the Pittsburgh Science of Learning Center which is funded by the National Science Foundation award number SBE-0836012. Thanks to Thomas Harris, Tristan Nixon, and Steve Ritter for their support concerning the use of the Carnegie Learning Cognitive Tutor Algebra code, and to Gail Kusbit, Christy McGuire, and the classroom teachers for their motivated involvement in the project. Finally, thanks to Carolyn Rosé, Dejana Diziol, Ido Roll, Ruth Wylie and Amy Ogan for their comments at various stages.

References

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.

Barab, S., & Squire, K. (2004). Design-based research: Putting a stake in the ground. *Journal of the Learning Sciences, 13*(1), 1–14.

Bernsen, N., Dybkjær, H., & Dybkjær, L. (1997). What should your speech system say? *Computer, 30*(12), 25–31. Dec. 1997.

Beyer, H., & Holtzblatt K. (1997). Contextual Design: A Customer-Centered Approach to Systems Designs. Academic Press.
Booth, J. L., & Koedinger, K. R. (2008). Key misconceptions in algebraic problem solving. In B. C. Love, K. McRae, & V. M. Sloutsky (Eds.), *Proceedings of the 30th Annual Conference of the Cognitive Science Society* (pp. 571–576). Austin: Cognitive Science Society.
Brusilovsky, P. (2001). Adaptive hypermedia. *User Modeling and User-Adapted Interaction, 11*(1/2), 111–127.
Brydon-Milier, M., Greenwood, D., & Maguire, P. (2003). Why action research? *Action Research, 1*(1), 9–28.
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.
Chaudhuri, S., Kumar, R., Howley, I., & Rosè, C. P. (2009). Engaging collaborative learners with helping agents. In V. Dimitrova, R. Mizoguchi, B. du Bulay, & A. Graesser (Eds.), *Proceedings of the 14th Intl. Conf. on Artificial Intelligence in Education (AIED 2009)* (pp. 365–372). Amsterdam: Ios Press.
Chi, M. T. H., DeLeeuw, N., Chiu, M.-H., & LaVancher, C. (1994). Eliciting self-explanations improves understanding. *Cognitive Science, 18*, 439–477.
Collins, A. (1999). The changing infrastructure of education research. In E. C. Lagemann & L. S. Shulman (Eds.), *Issues in education research: Problems and possibilities* (pp. 289–298). San Francisco: Jossey-Bass Publishers.
Constantino-Gonzalez, M., Suthers, D. D., & de los Santos, J. G. (2003). Coaching Web-based collaborative learning based on problem solution differences and participation. *International Journal of Artificial Intelligence In Education, 13, 2–4* (Apr. 2003), 263–299.
Corbett, A. T., & Anderson, J. R. (1995). Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction, 4*, 253–278.
Davidoff, S., Lee, M.K., Dey, A.K., & Zimmerman, J. (2007). Rapidly exploring application design through speed dating. In J. Krumm, G. D. Abowd, A. Seneviratrne, & T. Strang (Eds.), *UbiComp 2007: Ubiquitous Computing. Proceedings of the 9th International Conference, UbiComp 2007, Innsbruck, Austria, September 16–19, 2007* (pp. 429–446). Germany: Springer-Verlag.
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.
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.
Fischer, F., Mandl, H., Haake, J., & Kollar, I. (2007). *Scripting computer-supported collaborative learning—cognitive, computational, and educational perspectives. Computer-supported collaborative learning series*. New York: Springer.
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.
Hmelo-Silver, C. E. (2004). Problem-based learning: What and how do students learn? *Educational Psychology Review, 16*(3), 235–266.
Israel, J., & Aiken, R. (2007). Supporting collaborative learning with an intelligent web-based system. *International Journal of Artificial Intelligence in Education, 17*, 340.
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. (2009). 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 (pp. 897–964). The Educational Psychology Series*. New York: 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 web-based science learning at schools. In T. Koschmann, D. Suthers, & T.-W. Chan (Eds.), *Proceedings of the International Conference on Computer Support for Collaborative Learning 2005* (pp. 331–340). Mahwah: Lawrence Erlbaum Associates.

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, & Greer J. (Eds.), *Proceedings of Artificial Intelligence in Education* (pp. 383–390). IOS Press.

Kumar, R., Ai, H., Beuth, J. L., & Rosé, C. P. (2010). *Socially-capable conversational tutors can be effective in collaborative-learning situations*. Pittsburgh: Intl. Conf. on Intelligent Tutoring Systems.

Lazonder, A. W., Wilhelm, P., & Ootes, S. A. W. (2003). Using sentence openers to foster student interaction in computer-mediated learning environments. *Computers and Education, 41*, 291–308.

Leelawong, K., & Biswas, G. (2008). Designing learning by teaching agents: The Bettys Brain System. *International Journal of Artificial Intelligence in Education, 18*(3), 181208.

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.

McNamara, D. S., O'Reilly, T., Rowe, M., Boonthum, C., & Levinstein, I. B. (2007). iSTART: A web-based tutor that teaches self-explanation and metacognitive reading strategies. In D. S. McNamara (Ed.), *Reading Comprehension Strategies: Theories, Interventions, and Technologies* (pp. 397–421). Mahwah: Erlbaum.

Michaels, S., O'Connor, C., & Resnick, L. B. (2008). Deliberative discourse idealized and realized: Accountable talk in the classroom and in civic life. *Studies in the Philosophy of Education, 27*(4), 283–297.

Nicol, D. J., & Macfarlane-Dick, D. (2006). Formative assessment and self-regulated learning: A model and seven principles of good feedback practice. *Studies in Higher Education, 31*(2), 199–218.

Palincsar, A. S., & Brown, A. L. (1984). Reciprocal teaching of comprehension-fostering and comprehension-monitoring activities. *Cognition and Instruction, 2*, 117–175.

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.

Prichard, J. S., Stratford, R. J., & Bizo, L. A. (2006). Team-skills training enhances collaborative learning. *Learning and Instruction, 16*(3), 256–265.

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

Robinson, D. R., Schofield, J. W., & Steers-Wentzell, K. L. (2005). Peer and cross-age tutoring in math: Outcomes and their design implications. *Educational Psychology, 17*, 327–361.

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.

Rosé, C. P., & Torrey, C. (2005). Interactivity versus expectation: Eliciting learning oriented behavior with tutorial dialogue systems. In *Proceedings of Interact*, Springer Press.

Rosé, C., Wang, Y.-C., Cui, Y., Arguello, J., Stegmann, K., Weinberger, A., et al. (2008). Analyzing collaborative learning processes automatically: Exploiting the advances of computational linguistics in computer-supported collaborative learning. *International journal of computer-supported collaborative learning, 3*(3), 237–271. doi:10.1007/s11412-007-9034-0.

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

Saab, N., Van Joolingen, W. R., & Van Hout-Wolters, B. (2007). Supporting communication in a collaborative discovery learning environment: The effect of instruction. *Instructional Science, 35*, 73–98.

Schoenfeld, A. H. (1992). Learning to think mathematically: Problem-solving, metacognition, and sense making in mathematics. In D. Grouws (Ed.), *Handbook for research on mathematics teaching and learning* (pp. 334–370). New York: Macmillan.

Soller, A., Jermann, P., Mühlenbrock, M., & Martinez, A. (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*(4), 261–290.

Stahl, G. (2000). A model of collaborative knowledge building. In B. Fishman & S. O'Connor-Divelbiss (Eds.), *Fourth international conference of the learning sciences* (pp. 70–77). Mahwah: Erlbaum.

Stahl, G. (2009). Social practices of group cognition in virtual math teams. In S. Ludvigsen, Lund, A., & Säljö, R. (Eds.), *Learning in social practices*. ICT and new artifacts: Transformation of social and cultural practices. Pergamon.

Van den Bossche, P., Gijselaers, W., Segers, M., & Kirschner, P. (2006). Social and cognitive factors driving teamwork in collaborative learning environments. *Small Group Research, 37*, 490–521.

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

Vassileva, J., McCalla, G., & 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.

Walker, E., Rummel, N., & Koedinger, K. R. (2009). Integrating collaboration and intelligent tutoring data in the evaluation of a reciprocal peer tutoring environment. *Research and Practice in Technology Enhanced Learning, 4*(3), 221–251.

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.

Weinberger, A., Ertl, B., Fischer, F., & Mandl, H. (2005). Epistemic and social scripts in computer-supported collaborative learning. *Instructional Science, 33*(1), 1–30.