Educ Psychol Rev (2010) 22:89–102
DOI 10.1007/s10648-009-9116-9

REVIEW ARTICLE

Using Intelligent Tutor Technology to Implement Adaptive Support for Student Collaboration

Dejana Diziol · Erin Walker · Nikol Rummel · Kenneth R. Koedinger

Published online: 18 December 2009

Abstract Research on computer-supported collaborative learning has shown that students need support to benefit from collaborative activities. While classical collaboration scripts have been effective in providing such support, they have also been criticized for being coercive and not allowing students to self-regulate their learning. Adaptive collaboration support, which would provide students with assistance when and where they need it, is a possible solution. However, due to limitations of natural language processing, the development of adaptive support based on an analysis of *student dialogue* is difficult. To facilitate the implementation of adaptive collaboration support, we propose to leverage existing intelligent tutoring technology to provide support based on *student problem-solving actions*. The present paper gives two examples that demonstrate this approach and reports first experiences from the implementation of the systems in real classrooms. We conclude the paper with a discussion of possible future developments in adaptive collaboration support.

Keywords Collaboration scripts · Adaptive collaborative learning systems · Intelligent tutoring

While research on collaborative learning has generally shown that student interaction can increase group performance and individual learning outcomes, these positive effects are not always found (cf. meta-analysis by Lou *et al.* 2001). Often students show unequal engagement in the collaborative learning activity; a few group members take responsibility for the problem-solving, while others engage in *social loafing* and are not motivated to interact with their partners (e.g. O'Donnell 1999). Even if students are engaged in the interaction, they might not show the type of collaborative behaviour that is positively related to learning. For instance, students often answer a partner's question by merely telling

D. Diziol (✉) · N. Rummel
Institute of Psychology, University of Freiburg, Freiburg im Breisgau, Germany
e-mail: dejana.diziol@psychologie.uni-freiburg.de

E. Walker · K. R. Koedinger
Human-Computer Interaction Institute, Carnegie Mellon University, Pittsburgh, USA

them the correct solution. As Webb (1989) has shown, this behaviour can impede their partner's learning, while explaining the solution approach can improve it. Therefore, student collaboration needs to be supported in order to yield performance and learning benefits.

An approach that has been particularly successful in supporting collaboration is the use of a collaboration script, which structures student interaction to prompt fruitful collaborative behaviours that might not otherwise occur (e.g. Fischer *et al.* 2007). Classically, collaboration scripts have been implemented in a fixed manner, where the assistance provided to students does not change based on student behaviour. Implementing scripts adaptively such that the assistance given to students is tailored to their individual needs is a new and promising direction of research. However, this research is still at an early stage, as it is difficult to automatically assess student collaborative behaviour as it occurs. In this paper, we introduce an approach that promises to advance the development and implementation of adaptive collaboration support: leveraging intelligent tutoring models used in individual learning settings to support social interaction. We will present two examples from our own research and discuss possible future developments.

Fixed Collaboration Scripts

Several fixed script approaches have been effective in supporting student interaction and learning by guiding the collaborators through a sequence of interaction phases with designated activities and roles (cf. Kollar *et al.* 2006). For instance, jigsaw scripts structure student collaboration by *distributing the learning material* between interaction partners (Aronson *et al.* 1978; Dillenbourg and Jermann 2007). Students first review learning material in an “expert group” composed of group members that share the same task material, then each student is responsible for explaining his or her material to a “mixed group” that is composed of students with complementary knowledge. The knowledge distribution increases the individual accountability of each student and thus promotes student elaboration on the learning content and their overall participation. Other script approaches *assign roles* in order to improve student interaction (cf. Dillenbourg and Jermann 2007). For example, in the reciprocal teaching script by Palincsar and Brown (1984), students alternate between the roles of tutor and tutee. During their interaction, students engage in a sequence of elaborative activities involving summarizing, questioning, clarifying and predicting. By prompting these cognitive and metacognitive activities, reciprocal teaching effectively fosters student reading comprehension. Other script approaches *structure student dialogue* on a more fine-grained level by providing questions (e.g. “What would happen if ...”; King 1991) or sentence starters (e.g. “It was found that ...”; Kollar *et al.* 2005) that students have to apply during their collaboration. Sentence starters often foster particular content-related activities such as drawing conclusions or providing evidence and are sometimes referred to as “epistemic scripts” (cf. Weinberger *et al.* 2005).

Recently, many collaboration script approaches have been implemented in computer-supported settings (Kollar *et al.* 2006). By providing step by step instruction, these scripts lower the coordination costs that students typically experience when collaborating in computer-based settings. Furthermore, by embedding the script directly in the communication interface, the recommended activities can be made more salient or even compulsory. For instance, in Kollar *et al.* (2005), sentence starters were combined with pre-structured blank text boxes to enforce the construction of arguments by students. However, computer-mediated fixed script approaches also have their disadvantages. Fixed scripts, particularly when implemented in a computer environment, often provide overly coercive external control of the collaboration. This can reduce student motivation to interact—an issue that has been referred to as "overscripting" the interaction (Dillenbourg 2002). Students who are skilled at collaborating may already have fairly good internal collaboration scripts, thus they might not need so much external support (see Kollar *et al.* 2005). For students with little collaboration experience, on the other hand, weak interaction support will not produce the expected interactions. Providing adaptive collaboration support that responds to the individual needs of students may therefore be an improvement over fixed approaches (e.g. Dillenbourg and Tchounikine 2007; Rummel and Weinberger 2008).

Adaptive Collaboration Scripts

There is growing empirical evidence to suggest that adaptive script approaches may indeed be an improvement over fixed scripting. The effectiveness of adaptive support has primarily been evaluated using a Wizard of Oz approach (e.g. Gweon *et al.* 2006), where a confederate of the experimenter observes the interaction and gives adaptive support in pre-defined situations. Implemented in this manner, adaptive assistance has been shown to be better than no assistance (Gweon *et al.* 2006) and fixed assistance (Meier and Spada, in press) at increasing learning. However, this individualized support by a human is not feasible over the long term in a classroom environment. Thus, intelligent learning environments have been developed to provide automated support, and early evaluations of these systems have yielded positive results when compared to individual, unscripted or fixed script controls (Kumar *et al.* 2007; Baghaei *et al.* 2007).

In order to tailor collaboration support to student needs, a system has to be able to automatically assess student interactions and to provide adaptive feedback based on the difference between the assessment and a model of optimal collaboration (Soller *et al.* 2005). One way of assessing the quality of student interactions is by tracking student dialogue patterns, commonly accomplished by asking students to indicate the type of contribution that they are making before they compose it. For example, students may select a sentence starter like "We need to work together on this..." to begin their utterance. Based on the starters that students select, the system can make inferences about what students are saying, and use these inferences to provide feedback (Tedesco 2003). However, as students generally do not accurately label their utterances, the inferences that the system makes can be inaccurate. Thus, automated dialogue assessment solutions are beginning to be developed (Israel and Aiken 2007). So far, this technology has only been used successfully in limited ways, such as for classifying the topic of conversation (Kumar *et al.* 2007), or for assessing student accuracy when they use sentence starters (Israel and Aiken 2007). Some researchers try to circumvent the problems of assessing dialogue by relying on simple metrics like participation to trigger feedback. For instance, these systems evaluate the amount or length of contributions collaborators make to a shared workspace or to a dialogue and support the interaction by directly encouraging the non-contributors to participate more (Constantino-Gonzalez *et al.* 2003). Unfortunately, the same assessment metrics cannot be used to give students feedback on *how* to participate, which may ultimately be more valuable.

While the best way to assess collaborative interaction is yet an open question, intelligent tutoring technology that assesses individual problem solving to provide adaptive feedback is much farther along. Thus, using existing problem-solving models for individual learning to provide interaction support may accelerate the development of adaptive collaborative learning support. In intelligent tutoring, the system observes the student's problem-solving actions and compares them to a model of optimal student performance in order to provide feedback (VanLehn 2006). Intelligent tutoring systems have been shown to support individual learning in a variety of domains such as physics (VanLehn *et al.* 2005), mathematics (Koedinger *et al.* 1997) and reading (Beck *et al.* 2004). Some systems for adaptive collaboration support already partially capitalize on this technology by providing adaptive support towards *problem-solving*. For example, when students submit a group solution in COLLECT-UML (Baghaei *et al.* 2007), the system evaluates the solution using a constraint-based model, and provides feedback on the quality of the solution. Occasionally, this problem-solving support even leverages student talk rather than simply student action: When CycleTalk (Kumar *et al.* 2007) detects problem-relevant topics in student conversation, it engages the collaborating students in a tutorial dialogue, asking them to answer questions that concern these aspects. This tutorial dialogue often yields increased interaction between the collaboration partners. Both COLLECT-UML and CycleTalk realized support to a collaborative setting by extending an existing system that had been developed for individual learning. Similarly, in our research, we propose to use the problem-solving models of an existing intelligent tutoring system as input for the interaction model, in other words, we provide *interaction support* based on an evaluation of student problem-solving actions.

Using Intelligent Tutor Technology to Create Adaptive Collaboration Scripts

In the two systems we present in this paper, we used the output of existing problem-solving models (e.g. information on the correctness of a problem-solving step) as input to our models of interaction. In our studies, we worked with the Cognitive Tutor Algebra (CTA), a tutoring system for mathematics instruction on the high-school level that has been shown to increase student learning by approximately one standard deviation over traditional classroom instruction (Koedinger *et al.* 1997). The system covers different aspects of algebra learning such as linear equations and inequalities. Depending on the unit, students work with different tools that allow them to solve equations, plot equations in a coordinate grid or answer questions to word problems in a worksheet. To provide adaptive tutoring, the CTA evaluates the student's problem-solving actions by comparing them to a cognitive model of successful student performance, represented using a set of production rules. If an error is detected, the CTA immediately marks it as incorrect and provides context-sensitive feedback. In addition, students can actively request help from the CTA, receiving hints tailored to their current focus of attention.

We developed two collaborative extensions to the CTA. Both approaches provide adaptive support towards student interaction based on problem-solving cues. The first approach involves a collaborative problem-solving scenario. We use student collaborative problem-solving behaviour to directly infer when collaboration support should be provided. When the system detects ineffective learning strategies, it provides feedback to the dyad. The second approach involves a peer tutoring scenario. We use a combination of tutee problem-solving information and peer tutor actions to detect when peer tutors need support. In this section, we describe the implementation of the adaptive interaction support based on the cognitive models for problem-solving and illustrate their effects on student interaction when implemented in classroom environments.

Collaborative problem-solving in the Cognitive Tutor Algebra

In this project, we augmented the CTA with a script for collaborative problem-solving. While the CTA provided just-in-time feedback and on-demand hints in order to provide problem-solving support (as described above), the collaboration script aimed to support student interaction. The adaptive interaction support was embedded in a general script structure that followed a jigsaw-schema, in other words, it distributed expertise for solving the collaborative problem between the learning partners to set the stage for a fruitful interaction. During an individual phase, each student solved a different linear equation on the CTA; during a subsequent collaborative phase, students moved together on one computer to solve a system of equations problem that combined the two linear equations. Again, their problem-solving was supported by the CTA. Each problem was embedded in a real-world context (i.e. story problems). For instance, students compared two salary structures by answering several questions and comparing the corresponding graphs in a coordinate grid.

The *adaptive interaction support* assisted students' ongoing collaborative activities and helped them when impasses occurred. Adaptive interaction support was provided when the system detected ineffective learning strategies frequently used by students when learning in intelligent tutoring environments: trial and error and hint abuse (Baker *et al.* 2004). We detected these ineffective strategies based on problem-solving cues. First, students often misuse the CTA's error-flagging functionality by engaging in a trial and error strategy until they find the correct solution. This behaviour is negatively related to student learning (Baker *et al.* 2004), as students would better capitalize on the CTA feedback if they elaborated with their partner on how to correct errors. As the CTA's problem-solving model provides information on student errors, we were able to use this information to implement the adaptive interaction support. When students engaged in trial and error behaviour as indicated by multiple errors within a short-time interval, an adaptive script message encouraged student

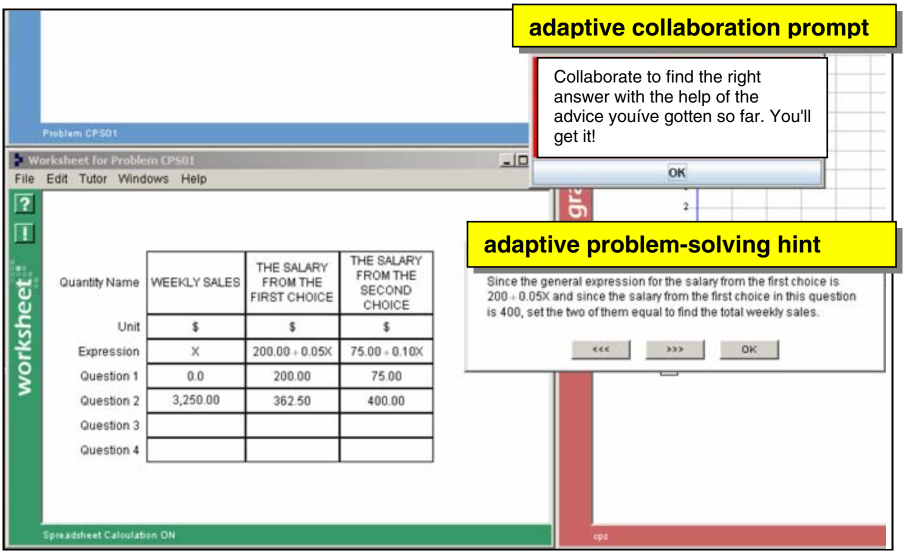

Fig. 1 Screenshot of adaptive hint prompt. In this system of equations problem, students compared two salary structures. Question 3 asked students to find the weekly sales necessary to earn US $400 from the first salary structure

interaction to yield deeper cognitive processing. Second, in the CTA, hints are presented in a hierarchical sequence with increasing level of detail. By making repeated help requests, students receive more and more detailed information and are finally provided with the correct answer in the "bottom-out hint" (Koedinger *et al.* 1997). Students often abuse the hints by clicking through the hint hierarchy and asking for the bottom-out hint, even though they might be able to find the answer on their own (Aleven *et al.* 2004). Based on students' hint behaviour that was logged by the problem-solving model, we detected their hint abuse. The interaction model then provided an adaptive prompt that encouraged the dyad to elaborate on the information given so far to find the answer on their own (see Fig. 1).

We implemented the adaptive collaboration script in several classrooms to evaluate experimentally if the script improved student interaction and learning when compared with students that collaborated on the CTA without interaction support (unscripted condition, see Diziol *et al.* 2007 for more information). The students were already familiar with the CTA environment. While we could not find differences between conditions regarding student learning outcome, the evaluation of student dialogue revealed that the adaptive collaboration reduced trial and error and hint abuse and increased student elaboration. Table 1 gives an excerpt from the dialogue of a scripted dyad that exemplifies how the adaptive interaction support influenced student collaboration. The dyad tried to solve the most difficult step of the system of equations problem: finding the intersection point. The adaptive collaboration support successfully intervened to prevent hint abuse and encouraged students to engage in mutual elaboration to find the solution on their own. In

Table 1 Dialogue Excerpt from Scripted Dyad

Activity	Talk	Analysis
Context of the dialogue: First system of equations problem. Students have to find out the amount of weekly sales for two salary structures to be equal		
	B: Just gonna ask for help then	Students do not yet know how to find the intersection point and thus ask for a hint
A requests a hint	A: OK, what do I do here?	
CTA hint: "*Given that the expression for the salary from the first choice and the salary from the second choice are equal, write an equation and solve it to find the total weekly sales.*"		The CTA launches a problem-solving hint
Dyad asks for following hint		The dyad asks for the bottom-out hint→hint abuse
CTA adaptive collaboration support: *"You've already gotten some advice. How can it help you to find the answer? Discuss with your partner."*		The system detects the hint abuse and provides adaptive collaboration support
A clicks OK to close the script window, clicks OK on the hint window		The adaptive prompt prevents student hint abuse. Starting from a solution idea of student B, the dyad partners subsequently engage in mutual elaboration on the help received and derive the equation that yields the intersection point on their own
	A: How do we do that?	
	B: Just enter the (.)...	

the second system of equations problem, the dyad no longer needed CTA assistance to solve this problem step.

In contrast, many dyads of the unscripted condition frequently engaged in ineffective learning strategies, indicating a need for interaction support. An example excerpt is given in Table 2. Even though this dyad had entered the interaction with a higher prior knowledge than the above-scripted dyad, they still did not know how to derive the intersection point when working on the third problem. This lack of learning can be explained by their ineffective collaboration behaviour: Every time they tried to find an intersection point, they deliberately engaged in trial and error and hint abuse. Thus, they did not learn how to perform this skill by the end of the study. Remarkably, scripted dyads transferred the improved interaction behaviour to subsequent interaction situations where script instruction was no longer available: In a future learning situation on the CTA (domain: inequalities), dyads of the scripted condition made significantly fewer errors than dyads of the unscripted condition. However, as the adaptive interaction support was embedded in the general jigsaw structure, we cannot conclude definitely whether this positive script effect was due to the adaptive collaboration support alone, or due to a combination of fixed and adaptive assistance. Furthermore, while the results of the interaction analyses are encouraging to extend this foray in adaptive collaboration support, the statistically non-significant differences in student learning outcomes make it clear that further research is needed to optimize the support.

Table 2 Dialogue Excerpt from Unscripted Dyad

Activity	Talk	Analysis
Context of the dialogue: In the first two problems, this dyad engaged in trial and error and hint abuse to find the intersection point. Now they are working on the third system of equations problem. They have to find out after how many working days the number of quarters in the coffee cans of Leana and David will be equal		
CTA hint: *"Given that the expression for the number of quarters in Leana's can and the number of coins in David's can are equal, write an equation and solve it to find the number of workdays from now."*		The CTA launches a problem-solving hint
	C: Oh, I thought that!	Student C might have an idea of how to solve the problem
D asks for following hint		Nevertheless, student D immediately asks for the bottom-out hint→hint abuse
CTA bottom-out hint**:** "Solve the equation $86 + 2D = 256 - 9D$"		
D moves the hint window to be able to see both the hint window and the CTA solver tool		Instead of elaborating on the help received, student D moves the hint window in order to facilitate copying the correct equation from the bottom-out hint. Student C dictates the equation without elaborating on the correct solution
	C: << dictates >> 86 plus 2D equals two 56 minus 9D	
D enters "$86 + 2D = 256 - 9D$" in solver tool		

Peer tutoring in the Cognitive Tutor Algebra

In the second project, we augmented the CTA with a *reciprocal peer tutoring* approach. Reciprocal peer tutoring has been successful at increasing learning in classroom settings, primarily because it encourages students to reflect, elaborate and feel more accountable for their knowledge (Roscoe and Chi 2007). We applied this approach to the domain of literal equation solving (e.g. problems like 'Solve for x, $ax + bx = c$') for first and second year Algebra students. In a preparation phase, students first prepared to tutor by solving problems individually using the CTA. In a collaboration phase, students took turns tutoring each other using the CTA interface. Students were put into pairs and alternated being tutors on sequential problems (e.g. on the first problem, Student A tutored Student B, and on the next problem, Student B tutored Student A). Tutees solved the problems as usual, and peer tutors marked problems right or wrong. Peer tutors and tutees could communicate with each other, asking questions and providing explanations, in an instant messenger window (see Fig. 2). Peer tutors had access to a worked-example solution to the problem.

As in the previous project, this fixed script structure was enhanced by *adaptive interaction support* that assisted students' ongoing collaboration, but in contrast to the previous project, the support was provided using an integrated problem-solving and interaction model. The tutoring system tracked the student actions and adaptively gave error feedback in two situations: First, when the peer tutor marked a tutee action correct that was actually incorrect; second, when the peer tutor marked a tutee action incorrect that was actually correct. In order to make this diagnosis, the tutoring system compared the output of the CTA assessment of a problem step (correct or incorrect) to the peer tutor action (correct or incorrect). In the case of a discrepancy, the assistance contained both a social prompt (e.g. "This step is wrong. Tell your partner what mistake they made. Here is a hint to help you.") and the cognitive tutoring support an individual learner would have received from the CTA. Ideally, this assistance helps the peer tutor to reflect on correct and incorrect problem-solving steps and ensures the correct feedback is given to the peer tutee. The tutoring system further provided hints on demand to the peer tutor, also by integrating the relevant CTA hint with an

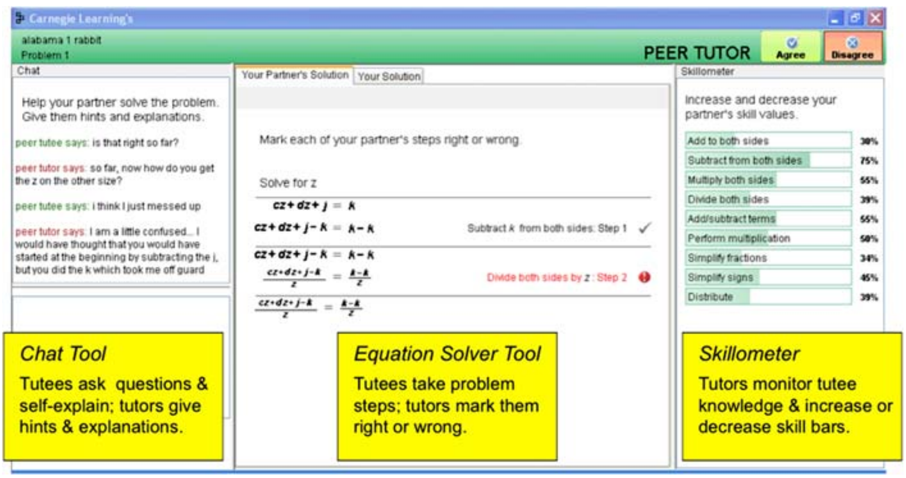

Fig. 2 Screenshot of the peer tutoring CTA environment

interaction prompt. This help on demand was designed to assist peer tutors in elaborating on their existing knowledge and constructing new knowledge. Students were prevented from moving to the next problem until they had completed the current problem.

In order to investigate the effects of this adaptive support on interaction and learning, we conducted a classroom study comparing a fixed support for peer tutoring condition (peer tutoring structure) and the adaptive support for peer tutoring condition described above (peer tutoring structure with adaptive collaborative tutoring support; see Walker *et al.*, in press, for more information). In both conditions, peer tutors had access to a correct worked example solution to the problems they were tutoring. We gave students pretests and delayed posttests to assess long-term retention. Students in all conditions learned between the pretest and the delayed posttest, but there was no significant difference between conditions, suggesting that in this case, adaptive and fixed support had similar effects on learning. Looking at how students collaborated in each condition and which collaborative behaviours related to learning, we found evidence that peer tutor learning was related to the reflective processes of tutoring, including viewing tutee errors, answering tutee help requests and

Table 3 Learning Opportunity Created by Tutor Feedback

Activity	Talk	Analysis
Context of the dialogue: The dyad was asked to solve the equation "$3q - xq = x$" for q		
Tutee selects "factor q," types "$3q=x$."		Tutee factors q incorrectly
Peer tutor approves "$3q=x$"		Peer tutor thinks the step is correct
	CTA (to peer tutor): Your partner has made a calculation mistake or skipped a problem step when typing in an answer. Please ask your partner to undo the incorrect steps and then redo the calculation	Peer tutor is made aware from the system that it is an error, creating a learning opportunity
	Peer tutor: "undo that step"	
Tutee divides by 3		
Tutee clicks done		
Peer tutor disagrees		Peer tutor understands that the tutee has not yet solved the problem
	Peer tutor: undo it	
	Tutee: why? U marked it right?	
	Peer tutor: the step is right but it said you made a typing error when you factored. *The dialogue continues until the tutee confirms which step to undo*	Peer tutor identifies the error for the tutee in an unelaborated way
Tutee undoes the step		
Peer tutor asks for a hint		
	CTA (to peer tutor): Your goal is to isolate q, but q is in all of the terms on the same side of the equation. Since q is a factor in more than one term, factor it out from each term	
	Peer tutor: Now factor out q. It should be $q(3-x)+x$. $q(3-x)=x$, sorry	Peer tutor tells the tutee how to complete the step, correcting his own error

giving yes or no feedback. The pattern of results did suggest that the adaptive condition had a positive influence on these reflective processes by making it easier for peer tutors to notice errors. However, the same errors were negatively correlated with tutee learning, and in general, the results suggested that peer tutors did not communicate in ways that might help tutees overcome their impasses. For example, peer tutors did not always communicate the adaptive feedback they received from the tutoring system to tutees, which led to confusion on the part of the tutee, and was negatively correlated with tutee learning. Table 3 presents a sample interaction from our study, where the peer tutor appeared to benefit from receiving computer feedback, realizing his own error, but did not articulate what he had learned to the tutee.

Conclusion

In this paper, we reviewed the recent progression from fixed to adaptive script support for collaborative learning. We pointed out the current difficulties in implementing adaptive collaboration support based on an assessment of student dialogue and proposed leveraging existing problem-solving models, often found in intelligent tutoring systems, to detect when the collaboration requires support. While adaptive collaboration support is still in its infancy, we believe that this approach can advance its development and implementation.

In this paper, we gave two examples of how interaction support based on problem-solving cues can be realized. In the first project, both students worked on one computer, and the CTA had a joint model of the dyad's problem-solving. The output of this problem-solving model served as input for an interaction model that assessed trial and error and hint abuse behaviours, triggering adaptive support that prompted fruitful collaboration. In the second project, one student tutored another student, located at separate computers. The system integrated a problem-solving model with a model of good tutoring in order to provide the students with support. There are several other design options for providing interaction support based on problem-solving models. While our systems modelled either one student (the peer tutor) or two students together, another option would be to model both interaction partners separately. For instance, the system COMET developed by Suebnukarn and Haddawy (2006) assesses the individual expertise of participants to encourage them to share their complementary knowledge with others at opportune times. Similarly, a system for adaptive collaboration support could intervene if it detects undesirable asymmetries in students' skill acquisition: If a particularly difficult problem-solving step is mainly solved by one interaction partner, this might indicate that the other student has not yet acquired the relevant skills. Adaptive collaboration support could encourage the weaker student to reflect on his or her partner’s solution procedure or to seek help from his or her partner while attempting to solve the step on his own.

Furthermore, leveraging existing problem-solving models can facilitate the comparison of different types of adaptive collaboration support, and thus provide us with information on the conditions of optimal assistance. While the evaluation of our systems revealed an impact on student interaction, the improved interaction did not yield the differences in student learning outcome that have been found in other studies (e.g. Baghaei *et al.* 2007). This null result indicates that further research on how to optimize collaboration support for particular interaction conditions might be necessary. This optimization can be considered an instantiation of a more general assistance dilemma (Koedinger and Aleven 2007), where in order to discover how to best deliver assistance, one must manipulate the amount, type and timing of help provided to students. For example, we so far only have limited knowledge on how to time support most effectively. It is an open question whether it is always best to provide adaptive collaboration support immediately, or whether it might sometimes also be beneficial to withhold it. Mathan and Koedinger (2005) investigated this research question for an individual learning setting, and found out that the two timing options served different goals. Immediate feedback ensured that students did not get stuck in problem-solving and thus was more immediately effective and efficient. Delayed feedback enabled students to practice their monitoring skills and consequently yielded improved learning transfer. Similarly, immediate feedback to collaboration may improve the current interaction, while delayed feedback may increase students' collaboration skills and thus promote future interactions (Kapur 2008). Thus, the type of feedback may have to be adapted to the goal of the instruction. On a practical level, the accelerated development of adaptive collaboration support conditions based on existing intelligent tutoring technology may help us to increase our knowledge of optimal collaboration assistance, as it enables us to more rapidly implement and compare different design options concerning the amount, type and timing of support.

Unfortunately, leveraging existing problem-solving models to provide support towards student interaction also poses limitations to the types of adaptive interaction support that can be implemented. On the other hand, systems that assess student dialogue automatically (e.g. Rosé *et al.* 2008)—while they may allow a larger variety of adaptive collaboration support—so far lack the necessary accuracy. As the output of the problem-solving models can provide information on the accuracy of the automated dialogue analysis, combining both approaches might be promising to push the field forward. For instance, in our current work, we automatically analyze student dialogue to assess the quality of the peer tutor's help (e.g. is it elaborated?) and combine this information with information from the problem-solving model (e.g. has the tutee recently made an error?) in order to decide when to provide assistance. We plan to formally assess whether the combined assessment metrics (interaction and problem-solving) are more effective than either one individually.

While there are first promising results that indicate the effectiveness of adaptive collaboration support, we have not yet developed sufficient knowledge to provide clear conditions and guidelines how to best deliver adaptive assistance. In this paper, we have shown that leveraging existing problem-solving models can facilitate the implementation of adaptive collaboration support. This approach can help us to investigate different types of adaptive collaboration support in more detail to increase our knowledge of when and why adaptive collaboration support is effective.

Commentary by Elisabeth Paus and Ina Jucks

Diziol, Walker, Rummel and Koedinger begin their article by outlining problems that typically emerge in collaborative learning scenarios. They introduce two common approaches for dealing with these issues: fixed and adaptive collaboration scripts. The authors describe several advantages of providing learners with fixed collaborative script support, but also criticize this traditional approach (as summarized in Fischer *et al.* 2007) with regard to its negative effects on interaction behaviour and learning. They argue that fixed support lacks flexibility and cannot be adequately adapted to learners' individual needs. Adaptive collaboration scripts, however, seem to be more promising. In order to adapt support in intelligent learning environments, the status quo of collaboration behaviour and learning has to be assessed automatically. Diziol and colleagues face this problem by using the output of existing problem-solving models as input for models of interaction.

They exemplify this approach by reporting on two of their own studies as first forays into the area of adaptive collaboration support, in which they combine collaboration scripts with intelligent tutor technology to adaptively support problem-solving in a collaborative setting. In the first study, learners communicated directly (face-to-face) while working at the computer. In the second study, social roles (tutor and tutee) were assigned to the learning partners, and only the tutor was supported by the CTA. Here, communication was solely computer-mediated. The studies showed that using problem-solving cues to infer the need for collaboration support is a promising approach to foster interaction and elaboration processes in collaborative learning environments: In the face-to-face setting, the introduced method assisted learners in using the help provided by the system more effectively; in the peer tutoring setting, the adaptive collaboration support helped the tutor to reflect on the tutoring process. On the other hand, both studies could not find clear results with regard to students' individual learning outcomes. One possible explanation might be that the adaptive collaboration support based on problem-solving was not sufficient, as it did not guarantee that students provided elaborated explanations to their partner. As a promising approach for future implementations of adaptive collaboration support, the authors propose to use the output of the problem-solving model to improve the accuracy of a dialogue analysis, so that positive behaviours such as providing elaborated explanations can be assessed and encouraged adaptively. However, it has to be considered that face-to-face interaction of partners in a dyad (study 1)—in comparison to fully computer-mediated communication (study 2)—makes it more difficult to assess the quality of interaction.

Dyadic or multiparty interaction? The computer (tool) as a communication partner?

Our comment approaches the work of Diziol and colleagues from the perspective of media and communication psychology. Drawing on works investigating human–computer interaction (Brennan and Clark 1996) as well as our own research into computer-mediated communication (Jucks *et al.* 2007), we suggest that the effects of the medium itself—in this case, the support software—should be considered as a component of collaboration support. It serves as an additional communication partner, influencing communication directly. The question arises how learners perceive the tool. Do they primarily focus on their own interaction and use the tool rather as support if they cannot solve a problem on their own? Or, do they perceive the tool as an infallible entity, which determines what is right or wrong, and defer to it in all cases? How much is natural collaborative behaviour influenced by assumptions (possibly shared between the learners) about effective tool use? These influences are not captured by objective measures. The authors' example of learners who misuse the help function illustrates this point very well: Learners deliberately engage in behaviour that is not conducive to learning in order to elicit the tool's goal-oriented help and support. We suggest integrating additional variables in further studies to evaluate the learners' subjective perception of the role of a software tool in the collaborative process.

These considerations lead to a second question: How much support should be given to the learners? The software tools introduced by the authors use problem-solving cues as a basis for adaptive collaboration support. This approach might hold the risk of denying students' agency. How can it be ensured that learners' naturally shown problem-solving and interaction behaviour is not disturbed by the system? And who is finally responsible for effective problem-solving and collaboration, the computer or the learning dyad?

Another drawback of oversupport is the demands placed on the person receiving the feedback, at the potential expense of other collaborators. Working with software can be particularly demanding for a peer tutor, who has to offer help and support to the tutee at the same time as monitoring and integrating the input from the system. As shown in the second study, the tutor is expected to guide and structure the learning process himself or herself. However, his or her attention is directed to the tool and his or her own learning process (i.e. rather than to the peer tutee's). He or she interacts with the system and benefits from this interaction, however, at the same time impairing learning of the tutee.

In summary, for further research we propose to consider the system as an additional agent. This can help us to better understand what type of interaction between the adaptive support tool and students is most beneficial for efficient collaborative behaviour and (thus) learning.

Acknowledgements This research was supported by the Pittsburgh Science of Learning Center, NSF Grant # 0354420, and by the Landesstiftung Baden-Württemberg. Images © 2009 Carnegie Learning.

References

Aleven, V., McLaren, B., Roll, I., & Koedinger, K. R. (2004). Toward tutoring help seeking: Applying cognitive modelling to meta-cognitive skills. In J. C. Lester, R. M. Vicari & F. Paraguaçu (Eds.), *Proceedings of Seventh International Conference on Intelligent Tutoring Systems, ITS 2004* (pp. 227–239). Berlin: Springer.

Aronson, E., Blaney, N., Sikes, J., Stephan, C., & Snapp, M. (1978). *The jigsaw classroom*. Beverly Hills: Sage.

Baghaei, N., Mitrovic, T., & 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.

Baker, R.S., Corbett, A.T., Koedinger, K.R. 2004. *Detecting student misuse of intelligent tutoring systems.* Paper presented at the 7th International Conference on Intelligent Tutoring Systems, Maceiò, Alagoas, Brazil, 30 August–3 September 2004.

Beck, J. E., Mostow, J., & Bey, J. (2004). Can automated questions scaffold children's reading comprehension? In J. C. Lester, R. M. Vicari & F. Paraguacu (Eds.), *Proceedings of 7th International Conference on Intelligent Tutoring Systems* (pp. 478–490). Berlin: Springer Verlag.

Brennan, S. E., & Clark, H. H. (1996). Conceptual pacts and lexical choice in conversation. *Journal of Experimental Psychology. Learning, Memory, and Cognition, 22*, 1482–1493.

Constantino-Gonzalez, M. A., Suthers, D., & Escamilla de los Santos, J. (2003). Coaching web-based collaborative learning based on problem solution differences and participation. *Artificial Intelligence in Education, 13*(2–4), 263–299.

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

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

Dillenbourg, P., & Tchounikine, P. (2007). Flexibility in macro-scripts for CSCL. *Journal of Computer Assisted Learning, 23*(1), 1–13.

Diziol, D., Rummel, N., Spada, H., & McLaren, B. (2007). Promoting learning in mathematics: Script support for collaborative problem solving with the Cognitive Tutor Algebra. In C. A. Chinn, G. Erkens & S. Puntambekar (Eds.), *Mice, minds and society. Proceedings of the Computer Supported Collaborative Learning (CSCL) Conference 2007, Vol 8, I* (pp. 39–41). International Society of the Learning Sciences.

Fischer, F., Kollar, I., Mandl, H., & Haake, J. M. (eds). (2007). *Scripting computer-supported collaborative learning—cognitive, computational, and educational perspectives*. New York: Springer.

Gweon, G., Rosé, C., Carey, R., Zaiss, Z. 2006. Providing support for adaptive scripting in an on-line collaborative learning environment. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 251–260). New York: ACM Press

Israel, J., & Aiken, R. (2007). Supporting collaborative learning with an intelligent web-based system. *International Journal of Artificial Intelligence and Education, 17*(1), 3–40.

Jucks, R., Bromme, R., & Runde, A. (2007). Explaining with non-shared illustrations: How they constrain explanations. *Learning and Instruction, 17*, 204–218.

Kapur, M. (2008). Productive failure. *Cognition and Instruction, 26*(3), 379–425.

King, A. (1991). Effects of training in strategic questioning on children's problem-solving performance. *Journal of Educational Psychology, 83*(3), 307–317.

Koedinger, K. R., & Aleven, V. (2007). Exploring the assistance dilemma in experiments with Cognitive Tutors. *Educational Psychology Review, 19*(3), 239–264.

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

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

Kollar, I., Fischer, F., & Hesse, F. W. (2006). Computer-supported collaboration scripts—a conceptual analysis. *Educational Review, 18*(2), 159–185.

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

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.

Mathan, S. A., & Koedinger, K. R. (2005). Fostering the intelligent novice: Learning from errors with metacognitive tutoring. *Educational Psychologist, 40*(4), 257–265.

Meier, A., & Spada, H. (in press). Developing adaptive collaboration support: The example of an effective training for collaborative inferences. *Educational Psychology Review.*

O'Donnell, A. M. (1999). Structuring dyadic interaction through scripted cooperation. In A. M. O'Donnell & A. King (Eds.), *Cognitive perspectives on peer learning* (pp. 179–196). Mahwah NJ: Lawrence Erlbaum Associates.

Palincsar, A. S., & Brown, A. L. (1984). Reciprocal teaching of comprehension-fostering and monitoring activities. *Cognition and Instruction, 1*(2), 117–175.

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., 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.

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

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

Suebnukarn, S., & Haddawy, P. (2006). Modeling individual and collaborative problem-solving in medical problem-based learning. *User Modeling and User-Adapted Interaction, 16*(3–4), 211–248.

Tedesco, P. (2003). MArCo: Building an artificial conflict mediator to support group planning interactions. *International Journal of Artificial Intelligence in Education, 13*, 117–155.

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

VanLehn, K., Lynch, C., Schulze, K., Shapiro, J. A., Shelby, R., Taylor, L., et al. (2005). The Andes physics tutoring system: Lessons learned. *International Journal of Artificial Intelligence and Education, 15*(3), 147–204.

Walker, E., Rummel, N., & Koedinger, K. (in press). Integrating collaboration and intelligent tutoring data in evaluation of a reciprocal peer tutoring environment. *Research and Practice in Technology Enhanced Learning*.

Webb, N. M. (1989). Peer interaction and learning in small groups. *International Journal of Education Research, 13*, 21–39.

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