Chapter 19

Helping Agents in VMT

Yue Cui, Rohit Kumar, Sourish Chaudhuri, Gahgene Gweon & Carolyn Penstein Rosé

YCui@cs.cmu.edu, RohitK@andrew.cmu.edu, Sourish@cmu.edu, GKG@cmu.edu, CPRose@cs.cmu.edu

Abstract: In this chapter we describe ongoing work towards enabling dynamic support for collaborative learning in the Virtual Math Teams (VMT) environment using state-of-the-art language technologies such as text classification and dialogue agents. The key research goal of our long-term partnership is to experimentally learn broadly applicable principles for supporting effective collaborative problem solving by using these technologies to elicit behavior such as reflection, help seeking, and help provision, which are productive for student learning in diverse groups. Our work so far has yielded an integrated system that makes technology for dynamic collaborative learning support—which has proved effective in earlier lab and classroom studies—available for experimental use within the "wild" VMT environment.

Keywords: Dialog agents, dynamic support, helping behavior, cognitive tutors, TagHelper, Basilica

Introduction

We are in the beginning stages of a long-term partnership, the goal of which is to enhance participation and learning in the Virtual Math Teams (VMT) online math service by designing, developing, implementing, testing, refining and deploying computer-based tools to support facilitation and collaborative learning in this lightly-staffed service. This project brings together the Drexel VMT Project—with the Math Forum's long track record and infrastructure for hosting and facilitating collaborative math problem-solving experiences in "the wild"—and the Carnegie Mellon team—

with expertise developing effective, state-of-the-art language technologies—to pursue the potential to create a new, more dynamic form of computer-supported collaborative learning than what has been possible in VMT until now. In addition to complementary technologies provided within the scope of this strategic partnership, insights from complementary methodologies come together in a powerful way. In this chapter, we describe our progress to date, both in terms of technological development and new insights gained from a "full circle" methodology, which takes insights from naturalistic observations, confirmed and refined through experimental lab studies, implemented within a technical infrastructure, and finally provided for future cycles combining naturalistic observations in the wild and refinement in controlled settings.

In the VMT environment, collaboration is currently supported with a combination of script-based support and human moderation. The script-based structuring is stage-based. Students typically work in small groups on the same problem over three or four sessions. In the first session, they work out solutions to the problem. In between the first and second sessions, students receive feedback on their solutions from human moderators. In the second session, students discuss the feedback they received on their respective solutions and step carefully through alternative correct solutions. In that session and the subsequent session, they also discuss additional possible ways of looking at the problem including variations on that problem in order to take a step back and learn larger mathematics principles that apply to classes of problems rather than individual problems. Although the problem provides the opportunity to investigate multiple possible solutions and to engage in deep mathematical reasoning, VMT researchers have found from analysis of chat logs where students have worked together that students tend to jump to finding one solution that works rather than taking the opportunity to search for alternative solutions. Prior work comparing students working with well defined versus non-specific problem-solving goals supports the belief that students can benefit from exploring multiple solutions, when those alternative solution paths provide opportunities to learn different concepts (see Chapter 10 for an example of re-use of problem-solving methods). Thus, there is reason to believe this typical pattern of narrowing to a single solution prematurely is counter-productive for learning. To address this and other issues, the moderator plays an important role in stimulating conversation between students, encouraging knowledge sharing and probing beyond a single acceptable solution.

While support from human moderators is extremely valuable to students, it is a rare commodity. Currently, only a tiny fraction of the approximately one million visitors to the Math Forum site each month have the opportunity to benefit from this expert-facilitated group-learning experience. Thus, our long-term goal is to greatly expand this capacity by using technology to support collaboration in this environment in two main ways, both of which leverage our prior research on automatic collaborative process analysis (Donmez et al., 2005; Rosé et al., 2008; Wang & Rosé, 2007). The first approach is to deploy *conversational agents* to offer fully automated support. As in our previous investigations in other collaborative environments, agents in VMT would participate in the student conversation.

Automatic analysis of the collaborative-learning process can be used to detect when a conversational agent should intervene in a conversation. Another direction we plan to pursue is to use the automatic analysis of the conversation to *construct reports* that inform human facilitators of which groups are most in need of support (Joshi & Rosé, 2007; Kang, 2008; Rosé et al., 2007). In this chapter, we focus primarily on the first approach.

The key research goal in the long term is to optimize a design and implementation for dynamic feedback in support of collaborative problem solving that will maximize the pedagogical effectiveness of the collaboration by eliciting behavior that is productive for student learning in collaborative contexts, including but not limited to the VMT environment. Towards this end, we have conducted a series of investigations across multiple age groups and multiple domains related to the design, implementation and evaluation of conversational agents that play a supportive role in collaborative-learning interactions (Chaudhuri et al., 2008; Gweon et al., 2006; Kumar, Rosé et al., 2007; Wang et al., 2007). We are working towards supporting collaboration in a dynamic way that is responsive to what is happening in the collaboration rather than operating in a "one size fits all" fashion, which is the case with state-of-the-art static forms of support such as assigning students to roles (Strijbos, 2004), providing prompts during collaboration (Weinberger, 2003), designing structured interfaces (e.g., with buttons associated with typical "conversation openings") (Baker & Lund, 1997), guiding learners with instructions to structure their collaboration (Webb & Farivar, 1999), or even various forms of training in collaboration (Rummel, Spada & Hauser, 2006). Our investigations thus far have been in lab and classroom studies. The far less controlled VMT environment provides a more challenging environment in which to test the generality and robustness of our prior findings, while at the same time providing a context where successful technology for supporting collaborative-learning interactions can reach a broad spectrum of students in need of support in their mathematics education.

While there has been much work evaluating a wide range of conversational agents for supporting individual learning with technology (Kumar et al., 2006; Rosé et al., 2001; Rosé & Torrey, 2005; Rosé & VanLehn, 2005; VanLehn et al., 2007), a similar effort in *collaborative* contexts is just beginning. We have observed in our recent research that working collaboratively may change the way students conceptualize a learning task and similarly how they respond to feedback (Wang & Rosé, 2007; Wang et al., 2007). For example, Wang & Rosé found that students approached an idea-generation task more broadly when they worked in pairs rather than as individuals, in particular behaving in a way that indicated more of a fluid boundary between tasks, whereas students who worked individually focused more narrowly on one task at a time. Correspondingly, students who worked in pairs with feedback showed even more evidence of a connection between tasks, where individuals with feedback during idea generation simply intensified their success within their original narrow focus. This difference in how students responded to feedback when they worked individually and in pairs tells us that before we will be able to effectively support collaborative learning—with tutorial dialogue technology (Gweon et al., 2005; Jordan et al., 2007; Rosé et al., 2001) in particular as well as intelligent

tutoring technology more generally—it will be essential to re-evaluate established approaches that have proven effective for individual learning now in collaborative contexts, and we have begun to engage in such comparisons (Gweon et al., 2007; Kumar, Rosé et al., 2007).

Our initial investigations using dialogue agent technology as collaborative learning support have been tremendously successful (Chaudhuri et al., 2008; Kumar, Gweon et al., 2007; Wang et al., 2007), suggesting that the presence of dialogue agents in the conversation increase learning of human participants as much (Kumar, Gweon et al., 2007) or more (Wang et al., 2007) than the human collaborators do.

We begin this chapter by describing our research methodology, which benefits from a combination of qualitative investigations conducted by Drexel's VMT team with insights from experimental studies and quantitative discourse analysis from our Carnegie Mellon team. Next, we describe our investigations of helping behavior, which are still in progress, but which have already suggested directions for dynamic support in the VMT environment. We will then describe our current integration between the technology for dynamic collaborative-learning support and the VMT environment, which we have now piloted with college-aged students in both math and thermodynamics (Chaudhuri et al., 2008). We will conclude with plans for future work.

Full-Circle Methodology: Complementary Insights from Complementary Contexts

In recent years, the CSCL community has grown in its openness to mixed methods and has made progress towards bridging a wide spectrum of methodological approaches from qualitative, ethnographic-style investigations to highly controlled, highly quantitative approaches. In that spirit, we leverage a broad spectrum of methodologies, ranging from high-internal-validity studies in the lab and in the classroom, with pre/post-test designs, to high-external-validity investigations in the "wild" VMT environment, where the same analyses of observable collaborative behavior are possible even with naturalistic, non-controlled observation, but experimental designs are less practical and must be administered with caution because of the way imposing too much control may interfere with the natural working of the community.

As an illustration of our full-circle, mixed-methods approach, we offer an example of how our informal collaboration to date is already yielding synergistic findings. This investigation provided the data for the quantitative investigation of math helping behavior discussed later in the chapter. Because our ultimate goal is to achieve success in the "wild" VMT environment, we begin with insights gained from an ethnomethodological analysis of chat logs collected in the VMT environment (see Chapter 5). In one notable chat session, the VMT team observed a group of students that was successful at solving problems collaboratively that none of them were capable of solving alone. On close inspection of the chat logs, a student who at first appeared as "the class clown" emerged as a tone setter in the analysis, putting his

team mates at ease, and allowing them to forge ahead as a group to solve a particularly challenging problem. From this analysis, a hypothesis emerges that interventions that inject humor in a collaborative-learning setting may act as a "social lubricant," and thereby may increase success in collaborative problem solving. The Carnegie Mellon team has tested this hypothesis experimentally in a classroom study in which students worked in pairs in a collaborative problem-solving environment that shares some common simple functionality with the VMT environment. We refer to this study as the Social Prompts study (Kumar, Rosé et al., 2007).

Experimental Infrastructure

The Social Prompts study was run as a classroom study with middle school students learning fraction arithmetic using a simple collaborative problem-solving environment (see Figure 19-1), which was a precursor to the integrated version of the VMT environment discussed later in the chapter. Although this study took place in a school computer lab, students worked in pairs, communicating only through text chat.

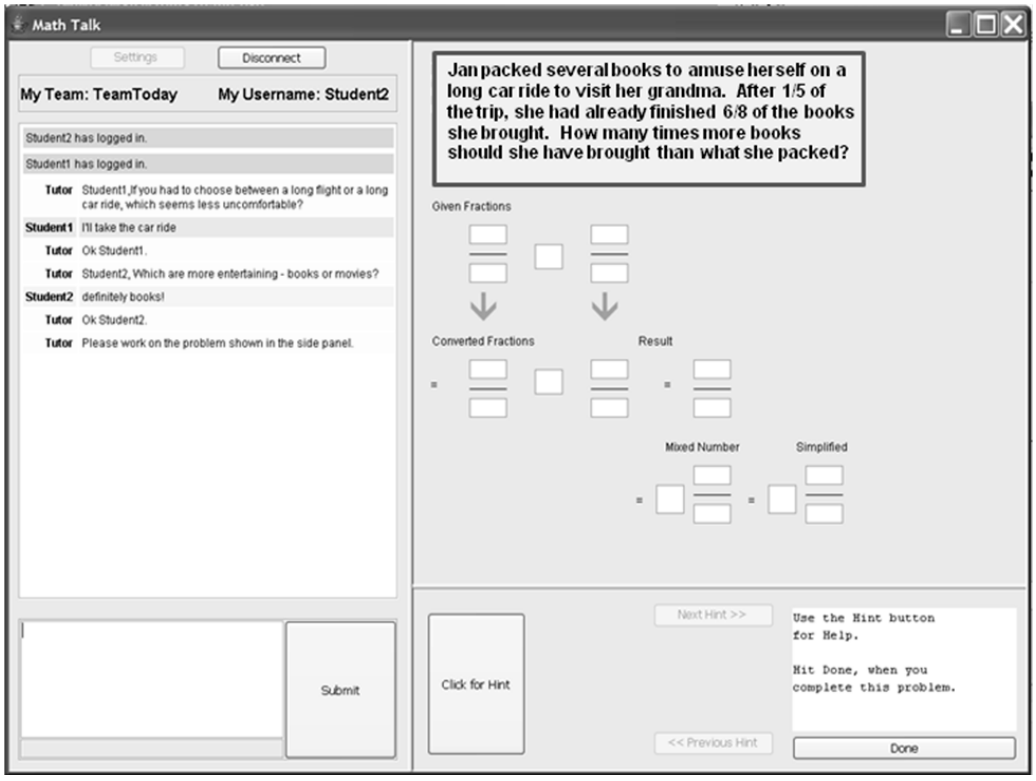

Figure 19-1. Early environment for collaborative math problem solving.

The interface in Figure 19-1 has two panels. On the left is a chat interface, which allows students to interact with each other as well as with conversational agents that

are triggered at different occasions during the problem-solving session to offer support to the collaborating pairs. The panel on the right is the problem-solving interface, which allows students to work collaboratively on a given problem. In this case the interface in the right panel was built using the Cognitive Tutor Authoring Tools (CTAT) (Aleven, McLaren & Koedinger, 2006). The problem-solving panel has a problem layout and a hint button. The hint button triggers support built into the environment. The hint messages are displayed in the chat window. Both panels of the interface maintain a common state across both the participants at all times, creating a shared experience for the student dyad. All actions performed by a student in either of the panels are immediately communicated and reflected on the interface of the other student. This integrated shared experience of problem solving is unique to this interface in contrast to systems used in our earlier experiments where VMT was used to manage the shared-problem-solving interaction (Gweon et al., 2007; Gweon et al., 2006).

Experiment and Results

The purpose of the experiment was to test the facilitative effect of off-task, social conversation on collaborative problem solving. Our hypothesis was that in a condition in which this form of social interaction was encouraged, students would work together better, offering each other more help, and thus benefiting more from the collaboration.

The experimental procedure extended over 4 school days, with the experimental manipulation taking place during days two (i.e., lab day 1) and three (i.e., lab day 2). The fourth day of the experiment was separated from the third day of the experiment by a weekend. Teams remained stable throughout the experiment. The students were instructed that the teams would compete for a small prize at the end of the study based on how much they learned and how many problems they were able to solve together correctly. The second and third days were lab days in which the students worked with their partner. Each lab session lasted for 45 minutes. At the end of each lab period, the students took a short quiz, which lasted about 10 minutes. At the end of the second lab day only, students additionally filled out a short questionnaire to assess their perceived help received, perceived help offered, and perceived benefit of the collaboration. On the fourth experiment day, which was two days after the last lab day, they took a posttest, which was used for the purpose of assessing retention of the material.

In the experimental condition of the Social Prompts study, before a problem is displayed in the shared problem-solving space, a tutor agent first asks each student what we refer to as a social question. For example, the agent may first ask student 1 "**Student 1, if you had to choose between a long flight or a long car ride, which seems more uncomfortable?**" The student indicates that a car ride would be preferable. Then the tutor agent may ask, "**Student 2, which are more entertaining—books or movies?**" The student may respond that books are more amusing. These two pieces of information are then used to fill in slots in a template that is used to generate personalized wording for the math problem. In particular, the resulting story problem says, "**Jan packed several books**

to amuse herself on a long car ride to visit her grandma. After 1/5 of the trip, she had already finished 6/8 of the books she brought. How many times more books should she have brought than what she packed?" The lighthearted nature of the word problem was meant to inject a note of humor into the conversation, and possibly spark off-task discussion, because the focus of the questions was on personal preferences of the students rather than strictly on math. In order to control for content and presentation of the math content across conditions, we used exactly the same problem templates in the control condition, but rather than presenting the social questions to the students, we randomly selected answers to the social questions "behind the scenes" Thus, students in both conditions worked through the same distribution of problems.

The results of the Social Prompts study provided some evidence in support of the hypothesis that emerged from observations in the VMT environment. We began our analysis by investigating the socially-oriented variables measured by means of a questionnaire, which we designed as a subjective assessment of perceived problem solving competence of self and partner, perceived benefit, perceived help received and perceived help provided. Each of 8 questions included on the questionnaire consisted of a statement such as "**The other student depended on me for information or help to solve problems**." and a 5 point scale ranging from 1 (labeled "strongly disagree") to 5 ("strongly agree"). For perceived benefit and perceived confidence, scores were high on average (about 4 out of 5) in both conditions, with no significant difference between conditions. However, with perceived help offered as well as perceived help received, there were significant differences between conditions (see Table 19-1). Students in the experimental condition rated themselves and their partner significantly higher on offering help than in the control condition. Interestingly, there is more evidence of requesting help in the control-condition chat logs. However, these requests were frequently ignored.

Table 19-1. Questionnaire results.

	Control	Experimental
Perceived Self Competence	4.2 (.56)	4.1 (.23)
Perceived Partner Competence	4.3 (.62)	3.9 (.49)
Perceived Benefit of Collaboration	4.5 (.74)	4.4 (.70)
Perceived Help Received	1.8 (1.3)	3.3 (.69)
Perceived Help Provided	1.8 (1.1)	3.1 (1.1)

The learning-gains analysis is consistent with the pattern observed on the questionnaire, and offers some weak evidence in favor of the experimental condition on learning. The trend was consistently in favor of the experimental condition across tests and across units of material on the test. The strongest effect we see is on lab day 2 where students in the experimental condition gained marginally more on interpretation problems ($p=0.06$, effect size 0.55 standard deviations). The student chat logs contain rich data on how the collaborative problem-solving process transpired.

We also conducted a qualitative analysis of the conversational data recorded in the chat logs in order to illuminate the findings from the test and questionnaire data discussed above. Overall, we observed that students were more competitive in the control condition. Insults like "**looser**," "**you stink**" or "**stupid**" occurred frequently in the control condition, but never in the experimental condition. Instead, in the experimental condition we observe light-hearted teasing. Furthermore, students referred to themselves as a group more frequently in the experimental condition. More details of the analysis of the chat logs are presented in the next section.

The full-circle methodology that we follow begins with ethnographic observations from interactions in the VMT environment. These observations lead to hypotheses that can be tested in high-internal-validity environments such as lab and classroom studies. These studies help us to confirm causal connections between actions and subsequent effects, between which we observe a correlation in our earlier ethnographic analyses. Discovered causal connections can then form the basis for the design of full-scale interventions, which can be prototyped and tested in the VMT environment. These investigations can eventually serve both as a test of the generality and robustness of findings from the lab and classroom studies as well as a source of new insights, forming the basis for new hypotheses that can be tested in further cycles—although only a large-scale controlled study evaluating the full intervention, such as we plan in the future, can provide definitive evidence of its effectiveness.

Analysis of Helping Behavior

Many of the benefits of collaborative learning are experienced through the discussions that students have in these contexts, so much of our work is focused on the dynamics of those conversations. For decades, a wide range of social and cognitive benefits have been extensively documented in connection with collaborative learning, which is mediated by conversational processes. The exchange of help is one valuable aspect of this process. Because of the importance of these conversational processes, in our evaluation of the design of conversational agents for supporting collaborative learning we must consider both the learning that occurs when individuals interact with these agents in the midst of the collaboration (i.e., learning from individual direct interaction with the agents) and learning that is mediated by the effects of the agents on the group interaction between the students. A formal analysis of helping behavior in our collected chat logs allows us to do that.

Theoretical Foundation

The help that students offer one another in the midst of collaborative learning ranges from unintentional help provided as a byproduct of other processes, to help offered with full intentionality. Beginning with unintentional help, based on Piaget's (1985) foundational work, one can argue that a major cognitive benefit of collaborative learning is that when students bring differing perspectives to a problem-solving situation, the interaction triggers consideration of questions and

ideas that might not have occurred to the students individually. This stimulus could help them to identify gaps in their understanding, which they would then be in a position to address. This type of cognitive conflict has the potential to lead to productive shifts in student understanding.

Related to this notion of cognitive conflict, other benefits of collaborative learning focus on the consequences of engaging in intentional teaching behaviors, especially the articulation of deep explanations (Webb, Nemer & Zuniga, 2002). Other work in the CSCL community demonstrates that interventions that enhance argumentative knowledge construction, in which students are encouraged to make their differences in opinion explicit in collaborative discussion, enhances the acquisition of multi-perspective knowledge (Fischer et al., 2002). Furthermore, based on Vygotsky's seminal work and his concept of the zone of proximal development (Vygotsky, 1930/1978), we know that when students who have different strengths and weaknesses work together, they can provide scaffolding for each other that allows them to solve problems that would be just beyond their reach if they were working alone. This makes it possible for them to participate in a wider range of hands-on learning experiences.

While the cognitive benefits of collaborative learning are valuable, they are not the only positive effect of collaborative learning. In fact the social benefits of collaborative learning may be even more valuable for fostering a productive classroom environment. These are obviously strongly related to the social interaction between students, which could be greatly enhanced by conversational interactions. By encouraging a sense of positive interdependence among students—where students see themselves both as offering help and as receiving needed help from others—collaborative learning has been used as a form of social engineering for addressing conflict in multi-ethnic, inner-city classrooms (Slavin, 1980). Some examples of documented social benefits of successful collaborative learning interactions include: increases in acceptance and liking of others from different backgrounds, identification with and commitment to participation in a learning community, improvements in motivation and aptitude towards long-term learning.

The social benefits of collaborative learning are closely connected with the Vygotskian foundations of collaborative learning because the positive interdependence that is fostered through collaborative learning is related to the exchange of support, or scaffolding, that students offer each other. Our own research has affirmed this connection. For example, in a previous study where we used a dynamic support intervention to encourage helping behavior, we observed anecdotal evidence that the manipulation increased helping behavior, and we observed a significant positive learning effect in the condition where we observed the increase in helping behavior (Gweon et al., 2006). In a subsequent study where we manipulated the availability of help from the problem-solving environment, we observed a significant positive correlation between the frequency of help offered and learning by the help provider (Gweon et al., 2006). In the same study, we observed that students perceived more benefit and learned more in the condition where they offered more help. Below we demonstrate through an analysis of the chat logs from the Social Prompts study introduced earlier that the presence of social dialogue

agents that show an interest in the personal preferences of participants not only created a more positive atmosphere between students and increased the perception of both help offered and help received, but also increased the concentration of actual verbal help exchanged per problem. As noted, the manipulation resulted in a marginal increase in learning on the second lab day of the study. All of these studies offer evidence of the value of helping behavior, consistent with what would be predicted by the theoretical foundations for collaborative learning put forward by Piaget, Vygotsky and others.

Simple Coding Scheme for Helping Behavior

In order to investigate whether students in the experimental condition actually offered each other more help in the Social Prompts study, we coded the chat logs from each lab day with a coding scheme developed in our previous work (Gweon et al., 2007). In order to make the sometimes cryptic statements of students clearer during our analysis, and also to provide an objective reference point for segmenting the dialogue into meaningful units, we merged the log-file data recorded by the problem-solving interface with the chat logs recorded from the chat window using time stamps for alignment. We then segmented the conversational data into episodes using the log files from the tutoring software as an objective guide. Each episode was meant to include conversation pertaining to a single problem-solving step as reified by the structured problem-solving interface. Between problems, conversation related to a single social prompt counted as one episode, and conversation related to one cognitive support agent also counted as one episode. All entries in the log files recorded by the tutoring software refer to the step in the action it is associated with as well as any hints or other feedback provided by the tutoring software. Note that steps where no conversation occurred did not have any episode associated with them in our analysis.

The simple coding scheme consisted of five mutually exclusive categories: (R) Requests Received, (P) Help Provision, (N) No Response, (C) Can't Help and (D) Deny Help. Along with the "other" category, which indicates that a contribution does not contain either help seeking or help providing behavior, these codes can be taken to be exhaustive.

The first type of conversational action we coded was Request Received (R). Help requests are conversational contributions such as asking for help on problem solving, asking an explicit question about the domain content, and expressing confusion or frustration. Not all questions were coded as Requests Received. For example, there were frequent episodes where students discussed coordination issues such as whether the other student wanted to go next, or if it was their turn, and these questions were not coded as help requests for the purpose of addressing our research questions.

Adjacent to each coded Request Received, in the column associated with the partner student, we coded four types of responses. Help Provisions (P) are actions that attempt to provide support or substantive information related to the other student's request, regardless of the quality of this information. These actions are attempts to move toward resolving the problem. Can't Help statements (C) are

responses where the other student indicates that he or she cannot provide help because he or she doesn't know what to do either. Deny Help (D) statements are where the other student responds in such a way that it is clear that he or she knows the answer but refuses to stop to help the other student. For example, "**Ask** [the teacher], **I understand it**" or "**Hold on** [and the other student proceeds to solve the problem and never comes back to answer the original question]" are type D statements. And finally, No Responses (N) are statements where the other student ignores help requests completely. Each chat log was coded separately by two coders, who then met and resolved all conflicts. Note that often where Requests Received are not met with a verbal Help Provision, the students are still able to collaboratively or independently work out an answer to their questions, at least at the level of moving forward with the problem solving. In some cases, however, the students seem to move forward through guessing.

Log 19-1 shows two example episodes where a Request Received is met with a Help Provision:

Log 19-1.

Student 1: What operation do we do?
<Student 2 tries multiplication and gets negative feedback from the problem-solving environment>
<Student 2 tries divide and gets positive feedback from the problem-solving environment>
Student 2: We divide. Now look at the problem, what is the other fraction we must divide by?

Student 1: What do we put on top of the fraction?
Student 2: Did you find a common denominator?
<Student 1 correctly finds the common denominator>

In Log 19-2 are two example episodes where a Request Received is met with a Can't Help response. In the second example, the student who requested help eventually figured out what to do on his own.

Log 19-2.

Student 1: Why 16?
Student 2: I don't know.

Student 1: I need help.
Student 2: Same
Student 1: 23/2
Student 2: What's 23/2?
Student 1: 11.5

Log 19-3 provides two example episodes where a Request Received is met with a Deny Help response. In the first case, the student who asked for help was able to figure out the answer by guessing.

Log 19-3.

Student 1: I don't get it
Student 2: hold on

<Then Student 1 tried something and got negative feedback from the problem-solving environment>
<Finally Student 1 tried something else, which was correct, and got positive feedback from the problem-solving environment>

Student 1: I don't know what to do
Student 2: click on the help button

Two example episodes where a Request Received is met with No Response are given in Log 19-4. In both cases the students seem to find the answer by guessing.

Log 19-4.

Student 1: I don't get it
<Student 2 tries something and gets negative feedback from the problem-solving environment>
<Student 2 tries something else and gets negative feedback from the problem-solving environment>
<Student 2 clicks on the help button>
<Student 1 tries something that is correct and gets positive feedback from the problem-solving environment>

Student 1: ?
<Student 2 tries something and gets negative feedback from the environment>
<Student 1 tries something, which is correct, and gets positive feedback from the environment>

The results from our coding of the corpus are displayed in Table 19-2. First, we see that there are a significantly larger total number of episodes on the transcripts from the Experimental condition. Recall that all episodes contain some conversation. Steps where no conversation occurred do not count in our total number of episodes. The larger number of episodes in the Experimental condition is primarily due to the fact that episodes in which social prompts were given to students only occurred in the Experimental condition, and two of these occurred between every problem solved during the Experimental condition.

Table 19-2. Results from corpus analysis.

	Experimental (Day 1)	**Experimental (Day 2)**	**Control (Day 1)**	**Control (Day 2)**
Total Episodes	47.1 (8.2)	61.3 (12.3)	33.8 (17.9)	49.1 (26.9)
Social Prompt Episodes	24.1 (9.9)	33.7 (16.2)	0 (0)	0 (0)
Solicited Help Episodes (P) ***Unsolicited Help Episodes***	.79 (1.6) *1.7 (2.1)*	.36 (1.1) *3.2 (6.0)*	1 (1.3) *2.1 (3.2)*	1.4 (2.9) *1.9 (3.2)*
Unanswered Help Requests (C+R+N)	2.4 (2.7)	1.4 (1.9)	2.2 (1.9)	1.4 (1.4)
Non-Help Episodes	19.9 (5.6)	35.8(9.3)	30.6 (16.3)	46.3 (25.1)

Looking at the totals in Table 19-2, our finding regarding the average number of Help Provisions was that—contrary to what we might suspect based on the questionnaire data—there was no significant difference between conditions, although there was a non-significant trend for fewer verbal Help Provisions to be given in the

Experimental condition. The number of Requests Received met with no verbal form of help was not different between conditions. However, there were significantly more non-help related conversational episodes in the control-condition transcripts. Furthermore, there were significantly more help episodes per problem in the Experimental condition $F(1,15) = 16.8$, $p < .001$, effect size 1 s.d. Thus, the students in the control condition may have perceived less help behavior because there was a lower proportion of helping behavior, both on a per problem basis (in terms of total amount of help per problem) as well as on an overall basis (in terms of proportion of conversational episodes altogether that were help related).

Ultimately it became clear to us that limiting our scope to verbal help was not adequate. As we examined both the verbal and non-verbal behavior of students, we began to see that sometimes where it appeared from the chat behavior that an explicit help request went unanswered, we saw behavior from the other student in the problem-solving logs that suggested that the student's intention was indeed to offer help, however not verbally. Thus, our recent work, discussed in the next section, has focused on characterizing help more broadly. Ultimately, we believe our efforts to monitor and support helping behavior in the VMT environment will need to account for both verbal and non-verbal behavior of students (e.g., through the VMT's whiteboard).

Extended Coding Scheme for Verbal and Nonverbal Helping Behavior

We are at the beginning stages of developing a coding scheme that captures both verbal and non-verbal helping behavior. This is a tricky analysis problem since from the non-verbal problem solving behavior we see it is often difficult to distinguish a case where a student is offering assistance non-verbally, from a case where a student is simply taking over the problem solving, and moving ahead without the partner.

From the collaborative problem-solving environment discussed earlier, log-files were generated that combined time-stamped records of each attempt to fill in a problem-solving step, along with who contributed that step and whether the contribution was correct or not, as well as time-stamped records of every contribution to the chat interface. We used the problem-solving behavior as an objective guide for segmenting the log-files into units for analysis. We used the problem-solving step as our unit of analysis. So all of the attempts to fill in a step counted together as one unit along with any chat contributions that were submitted during that time. Because the step was our unit of analysis, rather than coding each helping action as we had done in our earlier coding scheme, we coded each segment for what was the most explicit helping behavior, if any, we observed for each participant. Thus, at most we would assign one Request Received code and one Help Provision code per student, per segment. In what follows, we will describe the multi-step coding process.

The first step in the coding process is to mark for each step which student eventually entered the correct answer. This is used in the process of interpreting non-verbal help. We say that a non-verbal help request is initiated whenever the same student has contributed two unsuccessful attempts for the same step. The first time

this condition is true within a step, the student who is responsible for contributing that step is said to have non-verbally requested help by demonstrating a lack of ability. A code is then assigned that indicates how that help request was resolved, i.e., whether the student who initiated the help request was able to eventually contribute the right answer for the step without any intervention, either verbally or non-verbally from the other student, or whether the other student was the one who contributed the correct answer for the step, or whether the first student eventually contributed the right answer after receiving some form of help from his partner.

One interesting finding from this analysis was that between the experimental and control conditions of the Social Prompt study there were no significant differences in raw number of help requests, or number of help requests where the other student completed the step. However, between the experimental and control conditions there were marginally more cases in the experimental condition where a student requested help, received help from his partner, and then was able to complete the step himself using that help ($p = .07$).

Virtual Math Teams with Adaptive Support

In our recent work, we have integrated our technology for automatic analysis of the collaborative-learning process with our technology for supporting conversational interactions with computer agents into a single unified framework, which we refer to as Basilica. Conceptually, this framework allows monitoring conversational interactions as they unfold, which allows it to track behaviors such as argumentation, helping and explanation. Based on this analysis, conversational agents that are capable of engaging students in discussion on a variety of topics can be triggered. The purpose of this integrated framework is to make it possible to easily integrate these technologies with a wide range of collaborative environments. For example, we have recently integrated Basilica with the massively multi-player on-line environment called Second Life (Weusijana, Kumar & Rosé, 2008).

Example Interactions in the Integrated VMT-Basilica Environment

We have just begun to collect pilot data with the integrated VMT-Basilica environment using the types of open-ended math problems that are characteristic of prior investigations in the VMT environment. Here we describe our small pilot investigation in the math domain. A larger, experimental study in the thermodynamics domain is described elsewhere (Chaudhuri et al., 2008).

Figure 19-2 displays the interface for the integrated VMT-Basilica environment. It is identical to the original VMT environment, except that chat agents can participate along with a group of human students in the chat. (Note “Tutor” in the list of Current users and as the second poster in Figure 19-2.) On the whiteboard is displayed the problem statement the students are working on. Below it is a drawing one of the students has pasted on in order to share it with his partner. The goal of the exercise is for students to understand that they can use a binary search strategy to

solve the problem of finding a heavier weighted coin among a set of other coins that all weigh the same as each other.

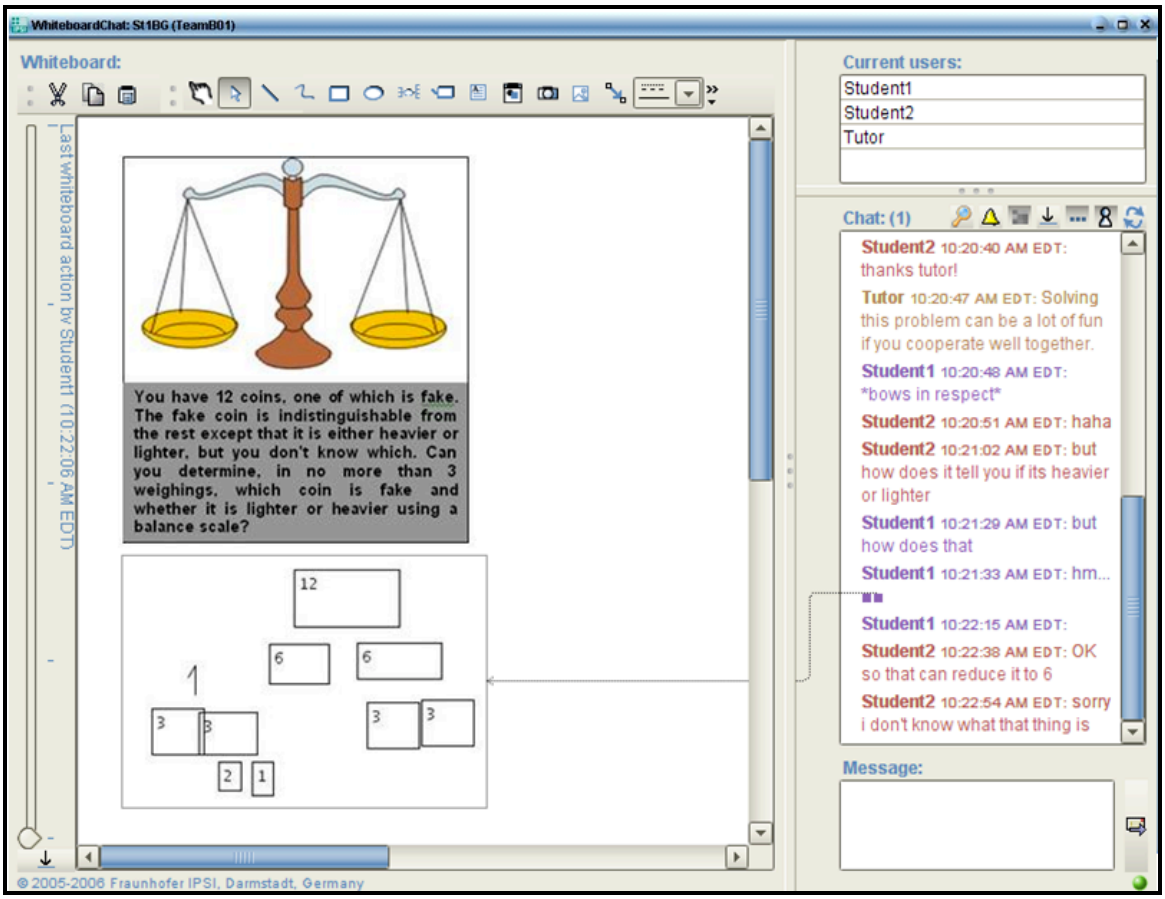

Figure 19-2. Integrated version of the VMT environment.

In Log 19-5, we see that the students first think together about a strategy for approaching the problem.

Log 19-5.

Student2: i was thinking what if you weighted them all out
Student2: and then took an average
Student2: but i guess you cant do that
Student2: because we cant know the weight
Student1: i was thinking of weighing them all, then weigh one (and hope it's not fake XP)
Student2: haha

After a short time, the tutor agent jumps in with a hint that the students then take up and use in their problem solving (see Log 19-6, from Figure 19-2).

Log 19-6.

Tutor: HINT: If you split a group of coins into half and they weigh equal, then that whole set doesn't contain the fake coin. It must be in the other set from the previous weighing. This also tells you whether the fake coin is heavier or lighter.
Student2: oh!
Student2: thanks tutor
Tutor: Solving this problem can be a lot of fun if you cooperate well together.
Student1: *bows in respect*
Student2: haha

Student 1 has worked out a solution based on the tutor's hint and then pasted it onto the whiteboard. This is indicated on Figure 19-2 with a small square in the chat following Student 1's "hm…" (see Log 19-5). When Student 2 clicks on this square, an arrow appears that points to what Student 1 has pasted onto the whiteboard. However, Student 2 is not able to fully understand Student 1's drawing (see Log 19-7).

Log 19-7.

Student2: but how does it tell you if its heavier or lighter
Student1: but how does that....
Student1: hm...
Student2: ok so that can reduce it to 6
Student2: sorry i dont know what that thing is
Student2: once we know its in one group of 6

After Student 2 has expressed confusion, Student 1 jumps in with a further explanation of the hint that originally had come from the Tutor agent (Log 19-8).

Log 19-8.

Student1: hm...so you recommend...first splitting the whole thing into groups of 3
Student2: yeah
Student1: then weighing two groups of 3 against each other
Student1: if they're equal, we use this as a benchmark against another group

The students were then able to move on from there to solve the problem. In addition to individual hints, the tutor agent is also able to engage students in multi-turn directed lines of reasoning, as in our previous investigations (Chaudhuri et al., 2008; Kumar, Gweon et al., 2007).

Technical Description: The Basilica Framework

Based on our experiences with designing and engineering collaborative-learning systems that involve integrating the state of the art in text classification and conversational-agent technology, we recognized the need for a framework that facilitates such integration. In this section, we describe our continuing work to develop such a framework. Initial specification of the desiderata of our framework included reusability of component technologies, compatibility with other platforms,

and the ability to provide flexibility to system designers to select from a wide range of existing components and then to synchronize, prioritize and coordinate them as desired in a convenient way.

While we continue to make further improvements to the framework to better achieve these specifications, in its current form the Basilica framework is an event-driven framework that enables development of conversational agents by using two basic components, referred to as Actors and Filters. These components communicate using Events. The Actor component, as the name suggests, displays behavior. Filters, on the other hand, observe behavior. Behavior and data are encapsulated into Events. For example if an Actor component generates a text message to be shared by the other participants in the conversational interface, it broadcasts a *TextMessageEvent*. Similarly, if a human logs into the conversational interface, a *LogInEvent* is sent to all relevant filters.

The Basilica framework implements a set of abstract software classes, which correspond to components, events and other supporting elements of the framework like channel independent communication, logging and process management. Along with these abstract classes, the Basilica framework now has a growing set of reusable Actors, Filters and Events that can be used to rapidly build custom conversational agents.

Supporting procedural behavior in an event-based framework like Basilica brings together two very different approaches of generating conversational behavior within the same framework. With this integration, we can view the conversational task as a space of graphs. Each graph represents a procedural behavior and graphs are triggered by events. The traversal of each graph once it is triggered is guided by the control strategy of the behavior associated with the graph. This has implications for the amount of effort involved in developing conversational applications of very different scales. In a procedural framework, the task may be represented by a simple graph or a quite complex graph. However if the task is designed appropriately, by using Basilica a complex graph can be divided into several smaller graphs, allowing distributed development with fewer dependencies and higher possibility of reuse of behavior. The event-driven approach adopted by Basilica has further advantages for building multi-modal conversational applications. This is particularly relevant if the different human participants engage in the conversation through different media like text, speech, short message service (SMS), gesture, etc. The programmatic approach to authoring taken by Basilica enables easier integration with devices and software for which a conversational interface has been developed.

A Basilica component can be defined on the basis of the events it observes and the operations it performs when it observes each of those events. These operations may include updating its beliefs, accessing its resources and generating events. Building a conversational agent under this framework is essentially an exercise in instantiating the desired actors and filters from a library of reusable components and establishing communication links, called Connections, between them. The exercise involves ensuring that all components are well connected to receive events they require. In some cases the exercise may involve creating new actors and/or filters to generate or observe some new behavior.

Figure 19-3 shows the design of Basilica within the integrated VMT-Basilica environment. Each Basilica component is represented by a polygon with three sections. The bottom section is marked with an out-pointing arrow on its left side and the text in that box is right aligned. The events generated by the component are listed in this box. The middle section is marked with an in-pointing arrow on its right side and text in this box is left aligned. The events received by the component are listed in this box. The name of the component is written in the top section. The shape of the top box determines the type of component it represents. An Actor component is drawn as a parallelogram and a Filter component is drawn as a rectangle. An "x" sign is placed in the corresponding box if a component does not receive or send any event.

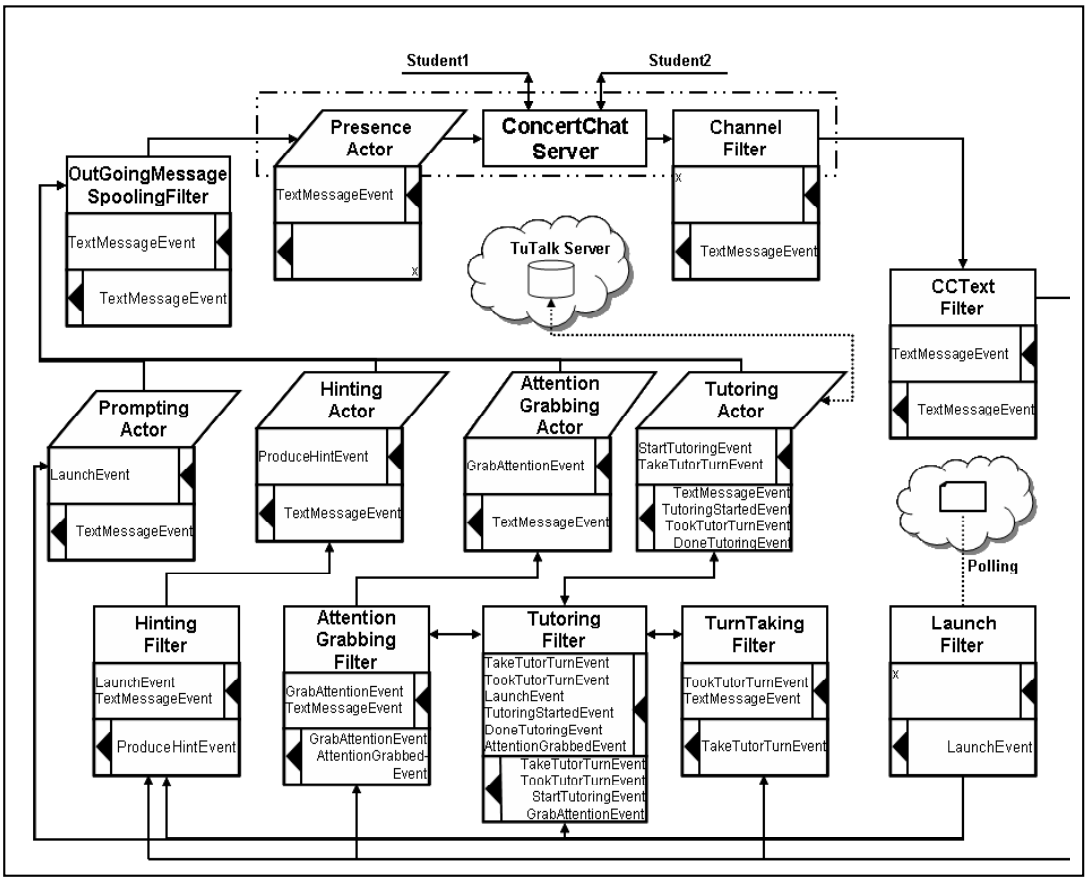

Figure 19-3. Configuration of Basilica.

In this integrated environment, the Basilica agent communicates with a VMT chat room (maintained by the ConcertChat server) using the *PresenceActor* and the *ChannelFilter*. The *PresenceActor* exhibits agent behavior like tutor turns, login and logout in the chat room while the *ChannelFilter* observes similar behavior by the students. All incoming text messages are encapsulated as *TextMessageEvent* and

routed to all Filters that need it through the *CCTextFilter*. On the output end, all outgoing messages are spooled through the *OutGoingMessageSpoolingFilter*.

There are four specific behaviors exhibited by the agent. These behaviors are distributed across four actors in the design. The *PromptingActor* prompts students at different times to inform them about the time left and also to give them certain motivating prompts at fixed time-points during the exercise. The *HintingActor* presents hints to the students to help them cover concepts that they have not yet discussed but which may be helpful in improving their design. The *HintingActor* works with the *HintingFilter,* which continuously monitors the students' conversation to evaluate which of the key concepts have been least discussed in the student conversations. Depending on that, the *HintingFilter* triggers the *HintingActor* to produce an appropriate hint at fixed time-points during the exercise. A hint is produced only once, and it is not reused during the exercise for the same students.

The *AttentionGrabbingActor* and the *TutoringActor* work together along with the *AttentionGrabbingFilter*, *TutoringFilter* and *TurnTakingFilter* to initiate and complete the instructional dialog session. At fixed time intervals during the exercise, the agent interacts with the students about certain concepts relevant for the exercise. Before an instructional dialog begins, to grab the student's attention, the tutor produces an *AttentionGrabbing* prompt. An example of an *AttentionGrabbing* prompt is "**Now might be a good time for some reflection**." The strategy to produce an *AttentionGrabbing* prompt is motivated from our past experience with building conversational support in a collaborative-learning environment. Students tend not to notice the tutor's instructional turn in between their turns, and the tutor never gains the floor in the conversation. We think that an *AttentionGrabbing* prompt may be successful at getting the tutor the floor in a text-based conversational environment.

Current Directions

In this chapter we have described our vision for enhancing the quality of support offered to students in the VMT environment. We began by discussing our mixed-methods approach to studying this problem and iteratively developing an evidence-based design. We have discussed our findings and continued explorations related to helping behavior as well as describing the current integrated environment that we have developed.

A major direction of our current work is to continue with our analysis of helping behavior. Ultimately, we would like to use TagHelper tools (Rosé et al., 2008) to automate this analysis, so that helping behavior in the VMT environment can be tracked and supported.

References

Aleven, V., Sewall, J., McLaren, B. M., & Koedinger, K. R. (2006). *Rapid authoring of intelligent tutors for real-world and experimental use.* Paper presented at the 6th IEEE

International Conference on Advanced Learning Technologies (ICALT 2006). Proceedings pp. 847-851.

Baker, M., & Lund, K. (1997). Promoting reflective interactions in a CSCL environment. *Journal of Computer Assisted Learning, 13*, 175-193.

Chaudhuri, S., Kumar, R., Joshi, M., Terrell, E., Higgs, F., Aleven, V., et al. (2008). *It's not easy being green: Supporting collaborative "green design" learning.* Paper presented at the *Intelligent Tutoring Systems (ITS '08)*.

Donmez, P., Rose, C., Stegmann, K., Weinberger, A., & Fischer, F. (2005). *Supporting CSCL with automatic corpus analysis technology.* Paper presented at the International Conference of Computer Support for Collaborative Learning (CSCL 2005), Taipei, Taiwan.

Fischer, F., Bruhn, J., Gräsel, C., & Mandl, H. (2002). Fostering collaborative knowledge construction with visualization tools. *Learning and Instruction. Special issue on measurement challenges in collaborative learning research, 12*, 213–232.

Gweon, G., Arguello, J., Pai, C., Carey, R., Zaiss, Z., & Rosé, C. P. (2005). *Towards a prototyping tool for behavior oriented authoring of conversational interfaces.* Paper presented at the *ACL Workshop on Educational Applications of NLP*.

Gweon, G., Rosé, C. P., Albright, E., & Cui, Y. (2007). *Evaluating the effect of feedback from a CSCL problem solving environment on learning, interaction, and perceived interdependence.* Paper presented at the *CSCL 2007*, Rutgers University.

Gweon, G., Rosé, C. P., Zaiss, Z., & Carey, R. (2006). *Providing support for adaptive scripting in an on-line collaborative learning environment.* Paper presented at the *CHI 06: ACM conference on human factors in computer systems*.: ACM Press.

Jordan, P., Hall, B., Ringenberg, M., Cui, Y., & Rosé, C. P. (2007). *Tools for authoring a dialogue agent that participates in learning studies.* Paper presented at the *AIED 2007*.

Joshi, M., & Rosé, C. P. (2007). *Using transactivity in conversation summarization of educational dialogue.* Paper presented at the *SLaTE Workshp on Speech and Language Technology in Education*.

Kang, M., Chaudhuri, S., Joshi, M., Rosé, C. P. (2008). *Side : The summarization integrated development environment.* Paper presented at the *Association for Computational Linguistics*.

Kumar, R., Gweon, G., Joshi, M., Cui, Y., & Rosé, C. P. (2007). *Supporting students working together on math with social dialogue.* Paper presented at the *SLaTE Workshop on Speech and Language Technology in Education*.

Kumar, R., Rosé, C. P., Aleven, V., Iglesias, A., & Robinson, A. (2006). *Evaluating the effectiveness of tutorial dialogue instruction in an exploratory learning context.* Paper presented at the *Intelligent Tutoring Systems Conference*.

Kumar, R., Rosé, C. P., Wang, Y. C., Joshi, M., & Robinson, A. (2007). *Tutorial dialogue as adaptive collaborative learning support.* Paper presented at the *AIED 2007*.

Piaget, J. (1985). *The equilibrium of cognitive structures: The central problem of intellectual development*: Chicago University 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 CSCL. *International Journal of Computer-Supported Collaborative Learning (ijCSCL), 3*(3), 237-272.

Rosé, C. P., Gweon, G., Arguello, J., Finger, S., Smailagic, A., & Siewiorek, D. (2007). *Towards an interactive assessment framework for engineering design learning.* Paper presented at the *ASME 2007 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*.

Rosé, C. P., Jordan, P., Ringenberg, M., Siler, S., VanLehn, K., & Weinstein, A. (2001). *Interactive conceptual tutoring in Atlas-Andes.* Paper presented at the *AI in Education*.

Rosé, C. P., & Torrey, C. (2005). *Interactivity versus expectation: Eliciting learning oriented behavior with tutorial dialogue systems.* Paper presented at the *Interact '05*.

Rosé, C. P., & VanLehn, K. (2005). An evaluation of a hybrid language understanding approach for robust selection of tutoring goals. *International Journal of AI in Education, 15*(4).

Rummel, N., Spada, H., & Hauser, S. (2006). *Learning to collaborate in a computer-mediated setting: Observing a model beats learning from being scripted.* Paper presented at the *Seventh International Conference of the Learning Sciences (ICLS 2006)*, Bloomsberg, IN: Lawrence Erlbaum Associates.

Slavin, R. (1980). Cooperative learning. *Review of Educational Research, 50*(2), 315-342.

Strijbos, J. W. (2004). *The effect of roles on computer supported collaborative learning.* Unpublished Dissertation, Ph. D., Open Universiteit Nederland, Heerlen, the Netherlands.

VanLehn, K., Graesser, A., Jackson, G. T., Jordan, P., Olney, A., & Rosé, C. P. (2007). Natural language tutoring: A comparison of human tutors, computer tutors, and text. *Cognitive Science, 31*(1), 3-52.

Vygotsky, L. (1930/1978). *Mind in society*. Cambridge, MA: Harvard University Press.

Wang, H. C., & Rosé, C. P. (2007). *Supporting collaborative idea generation: A closer look using statistical process analysis techniques.* Paper presented at the *AIED 2007*.

Wang, H. C., Rosé, C. P., Cui, Y., Chang, C. Y., Huang, C. C., & Li, T. Y. (2007). *Thinking hard together: The long and short of collaborative idea generation for scientific inquiry.* Paper presented at the *CSCL 2007*.

Webb, N., & Farivar, S. (1999). Developing productive group interaction. In O'Donnell & King (Eds.), *Cognitive perspectives on peer learning*: Lawrence Erlbaum Associates.

Webb, N., Nemer, K., & Zuniga, S. (2002). Short circuits or superconductors? Effects of group composition on high-achieving students' science assessment performance. *American Educational Research Journal, 39*(4), 943-989.

Weinberger, A. (2003). *Scripts for computer-supported collaborative learning: Effects of social and epistemic cooperation scripts on collaborative knowledge construction.* Unpublished Dissertation, Ph. D., University of Munich.

Weusijana, B. K., Kumar, R., & Rosé, C. P. (2008). *Multitalker: Building conversational agents in second life using basilica.* Paper presented at the *SLcc08: The Second Life Community Convention*, Tampa, FL.