Workshop on

Opportunities for intelligent and adaptive behavior in collaborative learning systems

Supplementary Proceedings of the Tenth
International Conference on Intelligent Tutoring Systems

Pittsburgh, PA, USA.
June 2010

http://grockit.com/blog/its-collaboration-workshop/

Preface

Intelligent tutoring systems are generally designed to tailor instruction to the individual student, but this does not mean that ITS-guided learning must necessarily be a solitary activity. A variety of recent systems have demonstrated ways in which an adaptive learning environment can incorporate and benefit from the presence of multiple learners. Similarly, students using computer-supported collaborative learning systems have been shown to benefit from the introduction of adaptive support that targets the collaboration. In this workshop, we invite discussion and seek to explore ways in which the combination of collaborative and intelligent aspects of a system can benefit the learner by creating a more productive learning environment.

Researchers face many challenges when working with collaborative intelligent learning systems. This workshop will be a venue for people to discuss lessons learned about the practical difficulties involved in implementing intelligent support for collaborative learning and evaluating it in a rigorous manner. We encourage participants to share findings and theories on how we can overcome the barriers to developing adaptive support for collaboration in order to achieve results that a traditional ITS may not be able to offer, such as increased motivation and social skills in addition to improved learning outcomes.

One goal of this workshop is for participants to leave with a new set of ideas surrounding techniques to consider (or avoid) when developing adaptive support for collaborative learning. In short, we wish to share knowledge about the unique challenges we face in building collaborative intelligent learning systems. What techniques have we found to be successful (or unsuccessful) in addressing these challenges? Why? And how do we know that these systems are worth all this effort?

Workshop contributors have approached the intersection of collaboration and adaptive support in different ways, such as the use of adaptive domain models to prompt collaborative discussion, the use of software agents to communicate directly with student collaborators in order to support their interactions, and the use of new methods to support asynchronous discussion. The workshop discussion will be focused on three sub-topics, chosen based on the interests of program committee members and contributing authors.

Modeling and Assessment This area focuses on broad questions related to the modeling and assessment of collaboration, such as: Which interactions do we want to encourage or discourage in collaborative learning systems? How can we assess the effectiveness of student interaction? This topic also includes questions about how to leverage some specific advantage of collaborative intelligent learning systems: How can a *group*'s understanding of the domain be modeled, in contrast to an *individual*'s understanding of the domain? What are the relative advantages and disadvantages of supporting *problem-solving* in a collaborative scenario, in comparison to focusing solely on *interaction*?

Collaborative Context This topic covers the advantages of taking a view of the collaborative context above the level of single actions of individual collaborators. Several granularities of support are possible in these systems, including the individual, the group, and the community as a whole. How can we incorporate the context in which the collaboration takes place into the support that an intelligent system provides? What roles can/should the computer play in relation to the participating students (e.g. tutor or learning companion)? Finally, how can we encourage the students in a collaboration to monitor and support their own interactions?

Scale and Sustainability New research opportunities arise as a collaborative system scales. Partner recommendation and group formation algorithms are examples of ways that an intelligent system can make collaborative learning more effective, and the value of these techniques increase with the number of participating students to consider. What other techniques improve with scale? Given the additional time and effort required to build support for collaboration into new intelligent learning systems, what practical lessons can we share to expedite the development process for future systems? How can we leverage existing architectures (either intelligent or collaborative) in building new systems?

Each paper that follows plays a dual role: it is both a stand-alone introduction to an ongoing research project and, within the context of the workshop itself, a shared marker serving to ground the discussion in real-world experiences, studies, and systems. We hope that you find these papers valuable in both of these roles.

June 2010

Ari Bader-Natal, Erin Walker, Carolyn P. Rosé
Workshop Chairs

Workshop Organization

Program Chairs

Ari Bader-Natal (Grockit, Inc.)
Erin Walker (Carnegie Mellon University)
Carolyn P. Rosé (Carnegie Mellon University)

Program Committee

Christa Asterhan (University of Pittsburgh)
Roger Azevedo (University of Memphis)
Nilufar Baghaei (Commonwealth Scientific & Industrial Research Organisation)
Bob Hausmann (Carnegie Learning Inc.)
Ulrich Hoppe (University of Duisburg)
Rose Luckin (University of London)
Bruce McLaren (DFKI – Saarbrucken / Carnegie Mellon University)
Helen Pain (University of Edinburgh)
Nikol Rummel (University of Freiburg)
Beverly Woolf (University of Massachusetts – Amherst)

Table of Contents

Assisting the facilitator: Striking a balance between intelligent and human support of computer-mediated discussions 1
Christa Asterhan, Baruch Schwarz

Combining peer-assistance and peer-assessment in a synchronous collaborative learning activity 5
Ari Bader-Natal

Promoting Collaboration and Discussion of Misconceptions Using Open Learner Models 9
Rasyidi Johan, Susan Bull

Modeling Successful vs. Unsuccessful Threaded Discussions 13
Jeonhyung Kang, Jihie Kim, Erin Shaw

Conversational Tutors with Rich Interactive Behaviors that support Collaborative Learning 17
Rohit Kumar, Carolyn Rose

Agents for Collaborative Learning in Virtual Worlds 21
Chunyan Miao, Han Yu, Zhiqi Shen, Xuehong Tao

Assessing, Modeling, and Supporting Helping Behaviors in Computer-Mediated Peer Tutoring 25
Erin Walker, Nikol Rummel, Ken Koedinger

Assisting the facilitator: Striking a balance between intelligent and human support of computer-mediated discussions

Christa S. C. Asterhan[1,2] & Baruch B. Schwarz[2]
[1]LRDC, University of Pittsburgh, USA
[2]The Hebrew University of Jerusalem, Israel
e-mails: asterhan@pitt.edu, msschwar@mscc.huji.ac.il

Research on computer-supported collaborative learning (CSCL) is traditionally rooted in constructivism. It intensively focuses on how productive peer collaboration can be stimulated and sustained in meticulously designed computer-mediated environments. Within the field of e-discussions, for example, it has been found that providing sentence openers, software-embedded collaboration scripts, representational guidance and even the medium itself may improve the quality of online collaborative argumentation. The role of the teacher and his/her impact on these activities, on the other hand, has been regarded to a far lesser extent. Teachers do not only plan, design and give feedback on students' collaborative activities, but they may also play an important role *during* these activities: They moderate, coach and guide groups of students. The research on F2F settings has unequivocally shown the positive effects of carefully calibrated, non-intrusive human facilitation of small-group discussions on its quality. When this is achieved in an on-line environment, it is often referred to as *e-moderation or e-facilitation*.

However, it is also known that e-moderation of group learning is not an easy task: Teachers not only have to monitor task progress and subject matter understanding, but the collaborative process as well. In an average-sized classroom or e-course with students working in small groups, the amount of information available to a teacher can become quite overwhelming. Compared to face-to-face group learning, this workload is even increased in CSCL environments (and especially synchronous discussion formats), since it lacks many of the traditional cues that teacher use to detect group dysfunction or individual difficulties. Moreover, since most CSCL environments are student-focused they do not offer tailored moderator tools that will allow teachers to unobtrusively intervene and support group work. Although e-moderation is a challenging task, CSCL environments also offer an opportunity to support e-moderation: Since many aspects of the collaborative process are logged, this information can be made available to teachers with the help of teacher-tailored visualizations of group interaction features (i.e., awareness tools), alerts and off-line analysis tools. Teachers can subsequently use this information to determine which activities or interventions they further need to initiate.

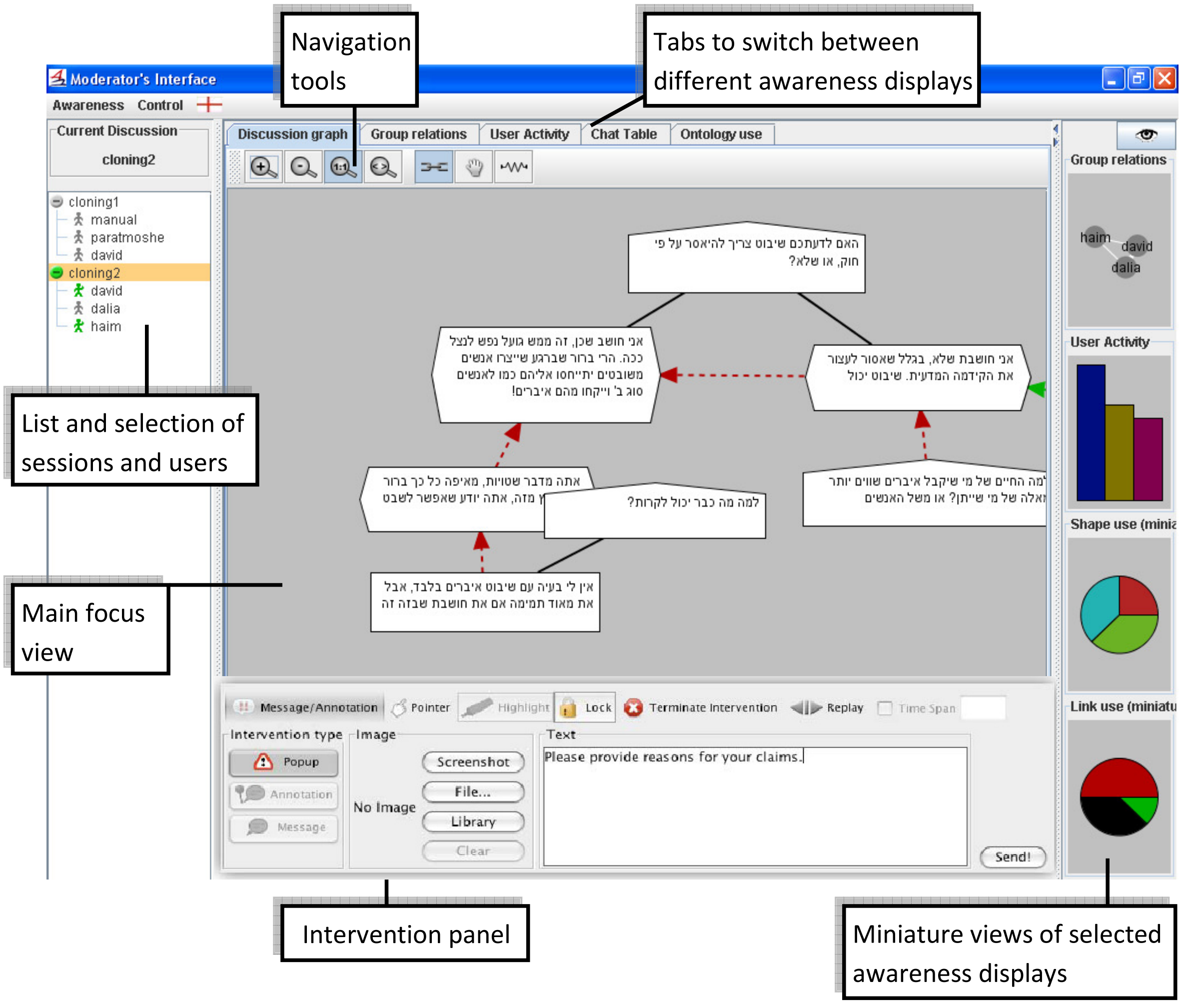
This combination of *intelligent* support for *human* facilitation of group processes seems to be particularly suited for ill-defined learning activities that involve multiple agents, such as small group discussions on social dilemmas and controversial issues. These activities do not have one (or even multiple) correct answers; the strength of a certain proposition or standpoint depends on the quality of the arguments brought forward to support it and to refute alternative views. In addition, such discussions often touch upon personal value systems and strong emotions. Guiding such discussions thus requires a deep understanding of rather complex group dynamics and subtleties. In the present paper, we describe a system that aims to capitalize on the combination of intelligent technology and on human expertise. This system, Argunaut, is designed to provide intelligent support for human facilitators of multiple discussions.

2 Description of the environment

The Argunaut system (De Groot et al, 2007; http://www.argunaut.org) is a platform, which combines two graphical discussion environments, among which Digalo (Asterhan & Schwarz, in press), a separate moderation environment and a module for user and session management. In this paper we refer to two of these components: (a) the Digalo v.2 discussion environment, in which students log in to and participate in pre-assigned discussion sessions through diagram-like representations (see Figure 1 for an impression); and (b) the Moderator's Interface, from which teachers or tutors can monitor these discussions and intervene when necessary. The Moderator's Interface (MI) is a multipurpose tool that can be used for real-time moderation of ongoing discussions as well as offline analysis of completed discussions. Despite these multiple uses, the main design goal was to generate a user interface for real-time moderation. It provides an interface capable of supporting simultaneous moderation of parallel discussions. It was designed in a collaborative, iterative design process involving pedagogical experts, technological experts, and teachers (Hoppe, de Groot & Hever, 2009).

The main user interface is a single window with a predefined layout. A typical view is shown in Figure 1. The window contains four main components: The session and user list (left column), the main focus view (center), remote control panel (bottom center, collapsed to a button), and aggregated miniature views (right column). We will shortly describe the first three:

Figure 1. Main window of Argunaut's Moderator's Interface.



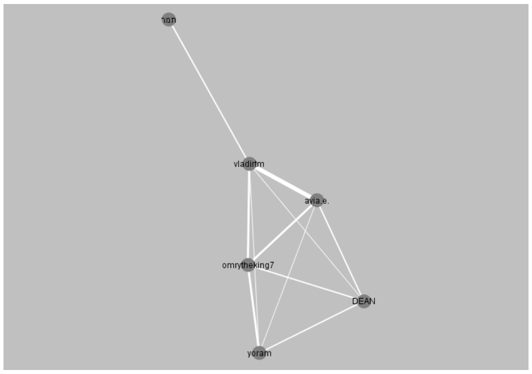
The *session and user list* includes tools for monitoring presence and for selecting groups and/or individuals within groups to be shown in the main focus view. Switching between different group discussions is done through this list. It is also responsible for showing alerts of important events in sessions other than the currently observed one. The alerting options that the MI offers range from the detection of superficial discussion features (based on keywords, inactivity, participation, responsiveness, etceteras) to alerts based on content-related dialogue analyses (e.g., patterns of reasoning, of interaction, see McLaren Scheuer & Mikšátko, in press). Since the alerting features were not activated in this study, we will not further report on them here.

The *main focus view* shows detailed information on the currently selected discussion with the help of a range of awareness displays that are continuously updated in real time. They are designed to provide quick and accurate updates on group and individual processes. Figure 2 presents four of the array of Awareness Displays moderators can choose from. By default, however, the main focus view shows the session's discussion graph, which is almost identical to the discussants' Digalo interface. Navigation through the main discussion graph enables the moderator to read the content of contribution (tooltip) and see how they are arranged. The moderator can resize and rearrange maps to follow the discussion as well as make patterns in the discussion appear clearer, all without affecting the discussants' environment.

The *Remote Control panel* enables real-time moderation of discussions (see bottom column in Figure 1). It offers a collection of tools to intervene in the discussion without actually being defined as one of the map's discussants and without acting from within the discussants' EUE. The moderator can choose to send these interventions to all groups, selected groups or (a) selected individual(s) only. This then enables both private and public communication, since the interventions are shown on the screens of selected users only. The three most relevant intervention options are (1) sending pop-ups with graphical and/or textual content; (2) attaching textual "stick-it" notes to one or more selected contribution shapes that are visually distinguishable from the discussants contributions; and (3) highlighting selected contributions.

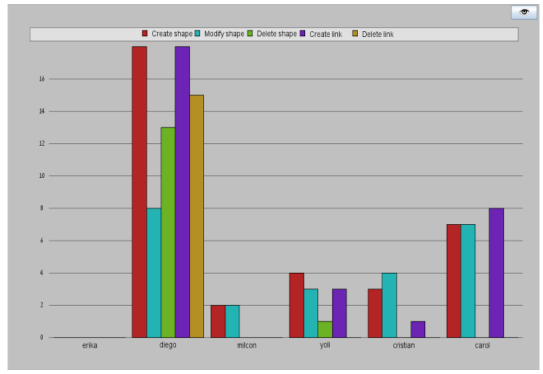
Figure 2. The four main Awareness Display tabs in the Moderator Interface.

(a) Group Relations



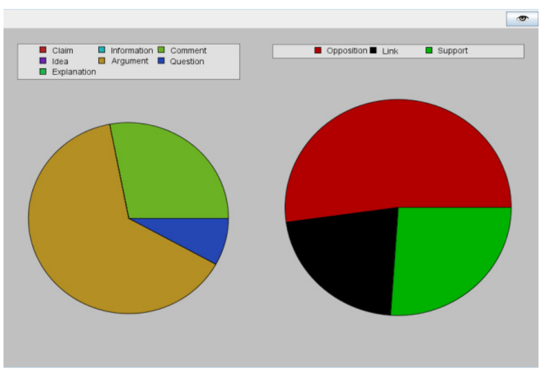
Each node represents a different discussant; width of links represents the frequency with which two discussants created links between each others' contributions (exact number visible with tooltips)

(b) User Activity



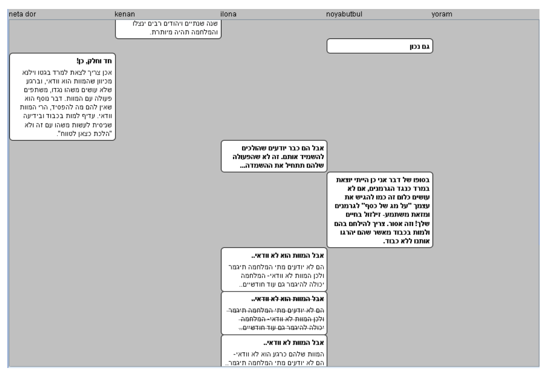
The x-axis shows nr. of activities. The y-axis shows name of participants and different bar colors represent different type of activities (e.g., create/delete/modify shape/link)

(c) Ontology Use



Pie charts show relative frequency of the use of different shape types (left chart; e.g., argument, question, explanation, claim) and different link types (right chart; e.g., neutral, opposing and supporting) in the discussion graph.

(d) Chat Table



Each column contains all the textual contributions from one discussant in a session. Contributions are vertically organized according to chronological order. Deletions or modifications are marked with the help of strike-through font and font colors.

3 A short description of the study

Twelve Israeli undergraduate students participated as discussants in this study. The discussions were not part of an existing course but were conducted in a co-located, laboratory type setting. One individual, Rhonna, moderated all discussions. She had mastered the technical aspect of Argunaut very well and had some, but not much prior experience with online moderation. Two moderated discussion sessions (1 two-group and 1 four-group session) were recorded with screen-recording software and converted to video-files. The topics for discussion discerned controversial topics: (1) the (dis)advantage of organized Holocaust Education trips for teenagers to Poland; and (2) whether the Gay Parade should be held in Jerusalem. These video files then displayed all the moderator actions and all the information received by the moderator within a given session. In addition, the actions of several discussants were videotaped with the same screen-recording technique. Two weeks following the experiment we interviewed Rhonna and several selected discussants separately, with the help of *cued retrospective reporting*, that is, they were asked to retrospectively comment on their actions while looking at the video file. Comments were audio-recorded in synchronization with the screen-recording files.

4 Selected findings and discussion

Unfortunately, space limitations do not allow us to present the complete and rich analyses of the way the moderator interacted with the participant s and moderated the discussion in a caring, yet non-intrusive manner, how she developed different moderation strategies and how this development was closely related to the affordances embedded in the software (see Schwarz & asterhan, in press, for a full report). We will show here only one illustrative example of how, on the one hand, the different MI features supported the human moderator in handling a particularly sensitive issue and how, on the other hand, the sensitivity, the empathy and understanding of complex social dynamics that a human expert can bring to the table was crucial in order to bring it to a successful solution.

Rhonna's initial strategy in both sessions was to observe the collaborative development of ideas and the contribution of each individual in it. This initial first strategy enabled Rhonna to notice particular behaviours that needed care. In one of the groups in the two-group session on educational trips to Poland's Holocaust landmarks, Rhonna noticed that one particular student had not contributed at all. She then realized that this particular student, Sohier, is a Christian Arab, whereas the other two and herself were Jewish. At first, Rhonna was not sure whether this student is not comfortable discussing the topic or whether she did not understand the question. To clarify this with Sohier, Rhonna used the private channel of the MI. From the clear position in favor of trips to Poland that Sohier expressed right after Rhonna's inquiry, she understood that the issue is not lack of understanding, but is socially motivated. She then had to switch to the other group, however, to monitor and support their progress. Upon returning to Sohier's group, her first care was to look at Sohier's engagement in the discussion by using the session and users session list to trace all her contributions and interactions in the discussion map. She quickly realized that Sohier still did not genuinely engage in the discussion and that Sohier found a way to respond to her (writing that she cannot follow up on the moderator's request). Rhonna immediately noticed this message and renewed her communication with her through the private channel. This time, she actively encouraged and supported her to participate. Among others, she stressed how her being different is actually an asset in the discussion, and carefully articulated a question that suggested how she could capitalize on her Palestinian identity to contribute to the discussion to the issue.

The third time that Rhonna attended to Sohier's engagement she used both the tracing options in the session and users list, as well as the Group Relations awareness display. She found that Sohier's vertex still appeared isolated from the other vertices, indicating little interaction between her and the other two discussants. However, she noticed that Sohier was in the midst of writing something and decided to await the content of her contribution before intervening. After a few minutes, she returned to Sohier's group and was pleased to find that Sohier had begun to express herself with a clear set of reasoned arguments. However, no one had reacted to her postings. In addition, and in line with the goal of the activity (critical, reasoned discussions) she would like Sohier to also consider alternative perspectives and be critical towards her own ideas. However, Rhonna hesitated on how to handle these two issues, since sending direct requests about either is likely to be interpreted as patronizing. She then browsed the Chat table for suitable postings by others that are relevant, but opposite to Sohiers', and used the highlighting and "stick-it note" functions of the Intervention panel to gently draw their attention to their potential connection, without further direct instructions.

This small sequence illustrates how an intelligent system can provide important information in a user-friendly way to support human moderators in their endeavors to facilitate several simultaneously running e-discussions. It also shows how in these ill-defined, complex, multi-agent activities human expertise and judgment is often called for to: (1) adequately evaluate the social and motivational dimensions of these complex interpersonal situations; (2) to flexibly and instantly adapt support for individual and group processes in ways that were foreseen or unforeseen; and (3) to intervene in a matter that is sensitive to these subtleties. The combination of intelligent support and human expertise then seems a promising combination.

References

1. Asterhan, C. S. C., & Schwarz, B. B. (in press). E-moderation of synchronous argumentative discussions: First findings from multiple sources. *International Journal of Computer-Supported Collaborative Learning*.
2. Hoppe, H. U., De Groot, R., & Hever, R. (2009). Implementing Technology-Facilitated Collaboration and Awareness in the Classroom: Roles for Teachers, Educational and Technology Researchers. In B. B. Schwarz, T. Dreyfus, & R. Hershkowitz (Eds.), *Transformation of Knowledge through Classroom Interaction: New Perspectives in Learning and Instruction* (pp. 130-142). New York, NY: Routledge.
3. McLaren, B. M., Scheuer, O., & Mikšátko, J. (in press). Supporting Collaborative Learning and e-Discussions Using Artificial Intelligence Techniques. *The International Journal of Artificial Intelligence in Education*.
4. Schwarz, B. B. & Asterhan, C. S. C. (in press). E-moderation of synchronous discussions i educational settings: A nascent practice. To appear in the *Journal of the Learning Sciences*.

Combining peer-assistance and peer-assessment in a synchronous collaborative learning activity

Ari Bader-Natal

Grockit, Inc.
San Francisco, CA USA
ari@grockit.com

Abstract. One unique characteristic of learning systems that support peer collaboration is that these systems have the potential to supplement or replace software-based representations of domain- and learner-models with the representations implicitly formed by peers. In order to realize this potential, a collaborative activity must sufficiently motivate peers to reflect, collect, and communicate these mental models. Peer-assessment represents a class of activities that address this challenge by design. In this work, we describe a project, currently under development, in which peer-assessment is melded with peer-instruction to create a new learning activity for an existing collaborative learning platform. We present the rationale behind the design of the activity, focusing specifically on how it draws from and synthesizes the three modes of learning supported by the Grockit platform: adaptive individual study, live collaborative small-group study, and instructor-led skill-focused lessons. By treating teaching as a demonstration of learning, we illustrate how a single activity can peer-assess mastery and peer-assist learning.

Keywords: collaboration, peer-tutoring, peer-review

1 Introduction

Part of the appeal of incorporating collaboration into an intelligent tutoring system is that collaboration introduces the available network of learners as a new resource to draw upon when designing and implementing the modeling and instruction processes. In letting students work together, these systems enable peers to serve as a new source of explanations, hints, and answers to unresolved questions and misunderstandings, which supplements the domain-specific intelligence built into the system with the natural intelligence of a student's cohort of peers. Additionally, the ability to interact with others introduces a social component into the learning experience that serves as a motivation for student engagement and retention [3]. However, just because students in these learning environments *can* engage in productive dialogue with their peers doesn't necessarily mean that they *will*. This poses a question to those who design collaborative learning systems, representing both a challenge and an opportunity: How can we affect the nature of student interactions by means of the design of the system itself? A variety of approaches have been pursued to create productive student collaborations, include (*a.*) leveraging game mechanics as a motivational structure to encourage specific types of engagement and interaction [1,2], (*b.*) determining appropriate opportunities during the course of a peer study session to prompt students (or some subset of them) with suggestions of a particular question to raise or topic to discuss [6], and (*c.*), defining specific roles, responsibilities, or scripts for each student to follow when participating [4].

In this work, we introduce a project that is currently in-development at Grockit, in which a new collaborative activity that combines peer-to-peer assessment and peer-to-peer instruction, which is embedded within Grockit's existing platform for synchronous collaborative learning. In this activity, a student can elect to reinforce knowledge and demonstrate mastery of a skill by leading a group of their peers through a sequence of challenges, taking on the role of a *peer-tutor* rather than simply a *peer*. This is tentatively called a *TeachIt* activity ("TeachIt to Grockit"). The outcome of the assessment is determined based on a combination of peer-evaluation, self-evaluation, and a quantitative metrics collected automatically by the ITS during the session. Heffernan and colleagues have built and studied tutoring systems that both assist and assess [5]. In this work, we seek to examine that combination in a collaborative context. In combining peer-assessment with peer-instruction, we believe that a system may be able to enhance learning through more productive collaborations and enable students to demonstrate – and be recognized by their peers for – mastery of specific skills in the domain.

2 Context: Grockit

As the TeachIt activity is situated within an established collaborative learning environment, a brief overview of the Grockit environment can both motivate and ground the design of the activity. Grockit (http://grockit.com) offers a web-based collaborative learning platform through which students can learn primarily through working practice problems, engaging in synchronous interactions with peers and with instructors, and by reading and asynchronously discussing expert-authored explanations. While the platform is currently being piloted in several school districts, most students use the system on their own time, such as high-school students continuing to study over the summer months or post-college students studying for graduate school entrance exams. For these students, Grockit offers a venue for studying with other students who share a common learning goal, which otherwise may not be feasible.

Three distinct modes of study are supported: (*a.*) individual practice, (*b.*) small peer-group study, and (*c.*) instructor-led lessons. The algorithms and affordances used in these three modes draw on three corresponding areas of research: (*a.*) Individual practice draws on work in the Intelligent Tutoring Systems field, including techniques for adaptively choosing challenges based on statistical models of response likelihood. (*b.*) Peer-group study draws on work on communicative activities in Computer-Supported Collaborative Learning, such as techniques for discussion scripting and group formation [4], and (*c.*) Instructor-led lessons draw on collaboration tools common in the E-Learning field, such as shared slides, whiteboards, real-time document editing, and audio streaming.

Each of these modes of study offers a different combination of benefits and drawbacks. Solo study offers the ability to target a study session to the specific needs of the individual student, but lacks the motivational effect of a social experience. Group study offers a scalable approach to creating an environment where students can raise questions and get immediate answers from others, but the quality of these collaborations may vary. Instructor-led lessons offers a structured environment in which one person leads the session, encourages discussion, and offers explanations and examples as needed, but a limited pool of instructors makes lessons difficult to coordinate and to scale. One of the goals for the TeachIt was to create a single activity that could draw on and combine these various benefits while avoiding or alleviating the associated drawbacks.

3 Concept: TeachIt

The TeachIt is a group session initiated and led by one student, as a way for that student to demonstrate their mastery of a particular skill in the learning domain. Domains focused on declarative knowledge and those focused on procedural knowledge seem to both be equally-suited for the activity. That student begins by selecting a skill for the TeachIt, among a list of options that may be limited based on some criteria (e.g. only skills that they have answered ten or more questions correct, only skills that they haven't already demonstrated mastery in, etc). The student can opt to begin the session immediately or schedule it for some time in the future (allowing others to plan to attend). When other students see the TeachIt in the list of joinable sessions, the special nature of the activity is communicated: Within the TeachIt, there will be one student responsible for leading discussion, answering questions raised by others, and explaining how to solve each problem. The session will not have the standard per-question timer (to allow for longer discussion). The TeachIt includes a fixed number of questions, all of which involve the specified skill. Finally, the student will be asked to complete a short peer-evaluation form following the conclusion of the session. For the student leading the TeachIt, expectations are also conveyed: their role in the session is explained, and they are told about the self- and peer-evaluations that will follow the session, which focus primarily on their ability to explain how to solve the problems as a demonstration of their own understanding, and their ability to address questions posed by their peers.

Following the conclusion of the activity, students complete brief evaluations based on a simple rubric. Free-text responses about the student's session are also elicited, and these are shared both with the student and with the community at-large. One possible direction to pursue with the design of the activity, currently under consideration, would be to make the record of each TeachIt publicly-accessible afterwards, effectively adding it to the student's public profile or participation portfolio.[1] Finally, quantitative data collected by the system itself may be taken into account, such as the number and difficulty of questions that the student answered correctly. The primary factor for determining the student's success will be the scores from the peer- and self-evaluation rubrics.

Classifying the TeachIt activity – with respect to the individual practice, small-group study, and instruct-led lessons – is not straightforward: The student leading the session chooses the specific topic for assessment/instruction, so the choice is based on that student's individual study needs at the time (a benefit generally associated with solo study). At the same time, the presence of other students in the session make it a social

[1] Doing so may motivate students to treat the assessment more seriously, or it may instead dissuade students from participating in the first place.

activity (a benefit generally associated with small-group study). For the other participating students, the TeachIt offers structured leadership approximating an instructor-led lesson, but without the scheduling restrictions inherent with a limited pool of available instructors. The TeachIt format illustrates how a collaborative activity can cross the boundaries of traditional modes of study, resulting in an experience in which peers play an active role in both teaching and testing one another.

The TeachIt format was the result of one set of responses to a set of high-level questions regarding how Grockit might assess mastery, such as: *Is assessment done by instructors or by peers? One-on-one or in group settings? Who initiates assessment? What are the evaluation criteria?* In using teaching as a demonstration of learning and in simultaneously making peer groups responsible for performing the assessment, we arrive at the design of a new activity that joins peer-assistance with peer-assessment in the context of an existing network of synchronous collaborative learning.

Acknowledgements

Many thanks to Farbood Nivi and the Grockit team for providing feedback and suggestions that continue to improve the TeachIt concept.

References

1. Ari Bader-Natal. *The Teacher's Dilemma: A game-based approach for motivating appropriate challenge among peers*. PhD thesis, Brandeis University, May 2008.
2. Ari Bader-Natal. Incorporating game mechanics into a network of online study groups. In Scotty D. Craig and Darina Dicheva, editors, *Supplementary Proceedings of the 14th International Conference on Artificial Intelligence in Education*, Intelligent Educational Games workshop, Brighton, UK, July 2009. IOS Press.
3. Ari Bader-Natal. Interaction synchronicity in web-based collaborative learning systems. In Theo Bastiaens, Jon Dron, and Cindy Xin, editors, *Proceedings of World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education 2009*, pages 1121–1129, Vancouver, Canada, October 2009. AACE.
4. Pierre Dillenbourg. Over-scripting CSCL: The risks of blending collaborative learning with instructional design. In A. Kirschner, editor, *Three worlds of CSCL: Can we support CSCL?*, pages 61–91. Heerlen: Open Universiteit Nederland, 2002.
5. Mingyu Feng, Neil T. Heffernan, and Kenneth R. Koedinger. Addressing the testing challenge with a web-based e-assessment system that tutors as it assesses. In *Proceedings of the 15th International World Wide Web Conference*, New York, 2006.
6. R. Kumar, C.P. Rosé, W. Yi-Chia, M. Joshi, and A. Robinson. Tutorial dialogue as adaptive collaborative learning support. *Artificial intelligence in education: building technology rich learning contexts that work*, 2007.

Promoting Collaboration and Discussion of Misconceptions Using Open Learner Models

Rasyidi JOHAN and Susan BULL
Electronic, Electrical and Computer Engineering, University of Birmingham, U.K.
rxj795@bham.ac.uk, s.bull@bham.ac.uk

Abstract. This paper describes the use of independent open learner models that prompt spontaneous collaboration amongst students, and suggest that these results could be built upon in systems that offer more explicit support for collaboration and student interaction. We focus in particular on the promotion of learner discussion around their misconceptions.

1 Introduction

Misconceptions have been investigated in a range of subjects, with the goal to recognise and understand learner 'knowledge' (e.g. [1],[2],[3],[4],[5],[6],[7]). While information about misconceptions is regarded as important to help teachers improve or target their teaching [8], information about misconceptions is also argued to benefit learners [9], and has been found useful in open learner model contexts [10].

Open learner models (OLM) are learner models that are 'opened' to the user in a form that they can understand. Common model presentations include skill meters and concept maps (e.g. [10]). Externalising the learner model can help prompt reflection and metacognition [11]. Previous research suggests learners are interested in knowing their misconceptions [10]. Furthermore, work also shows that where learners can optionally release their learner model to peers, spontaneous collaboration can be prompted, including discussion of misconceptions [12]. This finding was with an independent OLM: an OLM independent of a full intelligent tutoring system - there is no system guidance (for individual or collaborative learning), beyond presentation of the model to the user (and their peers, if applicable) [10]. Therefore, in this paper we consider whether an OLM could provide further support or a focus for collaboration in environments where collaborative interaction is explicitly facilitated.

2 Supporting Collaboration with Shared Open Learner Models

The potential for OLMs to prompt discussion of conceptual knowledge and misconceptions was suggested based on results of a pen-and-paper task with pairs of co-present students, after they had individually completed the same task [13]. The discussions (12 students / 6 pairs) demonstrated unprompted self-explanations and explanations to the learning partner; requests for explanations; statements of disagreement; self-questioning; and both collaboration and peer tutoring. The majority of students demonstrated increased understanding at the end of the experimental session.

A fully deployed independent OLM that indicates level of knowledge of a series of topics or concepts using skill meters, with specific statements of inferred misconceptions - e.g. *the '=' operator is used for comparison* (C programming) - has also demonstrated that students can be supported in collaborative face-to-face interactions even though it offers no explicit support for collaboration [12]. Students can choose whether to release their learner model to some or all peers in their course; whether to release this information named or anonymously; and whether to release it in the same way (named or anonymously) to different users. It was found that students did release their learner models to each other; that they undertook spontaneous face-to-face collaborative discussions based on the contents of their respective learner models; and that such discussion often focused on resolving misconceptions. This included interactions amongst students who did not usually discuss their work with each other [12].

OLMs have also been used to support teams of students working together on group projects, allowing students to reflect on how group members are contributing to the project [14]; and in distance learning to help overcome the feeling of isolation, and to allow students to compare their progress to that of others [15].

3 Step-By-Step Presentation of Misconceptions

AniMis uses the OLMlets [10],[12] learner modelling process (which builds a simple weighted model from multiple choice questions and permits viewing the OLM at any point during an interaction), but provides additional detail in the OLM. It allows students to explore step-by-step descriptions of their misconceptions and compare these against step-by-step descriptions of the corresponding correct concepts, and has so far been implemented for C programming and chemistry. In each case, descriptions are available in text and animation, as in Figure 1. The top left of Figure 1 is a step-by-step description of a concept; top centre is a simulation of the execution of a do-while loop; top right is an animation of a misconception. The bottom left shows a text description of a concept in chemistry; the lower middle gives an animation of a concept (simulating the chemical reactions taking place inside the cell, causing the flow of electrons and producing electricity). On the bottom right is an animation of a misconception (showing the electrons travelling through the solution to complete the circuit).

An evaluation with 14 users was undertaken with the programming version of AniMis during a one hour lab. Two users had no misconceptions, so this study uses data from the 12 participants who held at least one misconception. Participants were from Electronic, Electrical and Computer Engineering, University of Birmingham, UK. They were taking an Adaptive Learning Environments course, and had previously completed a C programming course. Log data and questionnaires were analysed, and the lab session was observed by a researcher. In this study, the feature to release the learner model to peers in OLMlets (see [12]) was not available.

The logs reveal 134 OLM viewings. All users accessed their model, with 2/3 viewing it more than 10 times (mean 17.625, median 11, range 2-31). Questionnaires indicate a majority of users found the OLM beneficial, but animations more useful (10 of the 12 users for misconceptions, 11 for concepts, no negative responses) than text (7 for misconceptions, 9 for concepts, 1 negative response - for concepts only). Although not instructed or requested to do so, students were observed to spontaneously discuss their understanding and learner models with each other. Typical exchanges included asking for explanations when misconceptions were identified; and more knowledgeable students spontaneously offering explanations or tutoring. In particular, some of the weaker students thought their misconceptions correct because the code output generated matched the correct answer, but then recognised the explanations of stronger students. These discussions typically involved 2-3 students who formed a group, mostly staying together for the session. Only 2 (of the 14 present) worked alone. On very few occasions did students ask for the correct answer, indicating a willingness to engage in collaboration and learning. Most discussion focused around animations.

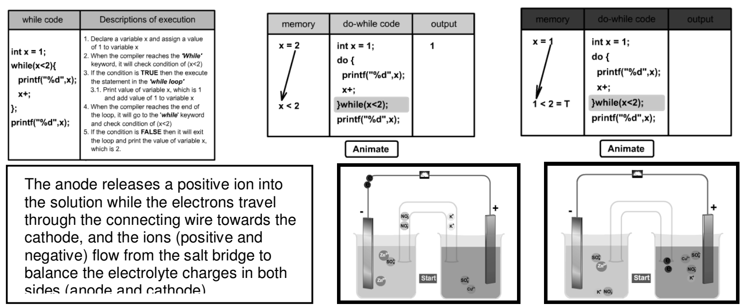


Fig. 1. Learner model views for C programming (upper) and chemistry (lower)

4 Summary and Conclusions

We have described an example of promoting spontaneous face-to-face discussion of misconceptions using OLMs. Previous work demonstrated face-to-face collaboration prompted simply by allowing students to release their models to each other [12]. The current study showed that even when users could not release their models to peers, they still came together spontaneously to discuss their OLMs, despite no instruction or suggestion that they should do so. Students claimed to find the OLM representations helpful. We therefore suggest that OLMs could be useful in prompting collaboration and discussion of misconceptions in a range of contexts - where OLM presentations are simple (e.g. [12]) or detailed (AniMis); where the models can be shared, or not released to others. In addition to the independent OLMs introduced above, this approach may also be useful in adaptive learning environments that explicitly facilitate collaboration, as the principle of providing a starting point for learners to collaborate and discuss their knowledge, still applies. Presenting learners with their inferred misconceptions and encouraging them to discuss these with each other, could be a powerful focus for the design of collaboration support. For example, concepts and (possibly multiple) corresponding misconceptions amongst learners, could be presented in a group model for online or face-to-face discussion, with system prompting as appropriate for the specific collaborative learning environment. Alternatively, system guidance for pairs/groups based on the kind of spontaneous learner groupings observed here, using the individual models of those participating, could be offered. Given the range of subjects in which misconceptions are found, we anticipate the approach to be broadly applicable. Future work will consider such issues with reference also to the nature of the misconceptions held and conceptual change (see [16]).

References

1. Brookes, D.T & Etkina, E. Using Conceptual Metaphor and Functional Grammar to Explore How Language Used in Physics Affects Student Learning, Physical Review Special Topics - Physics Education Research 3(1) (2007).
2. Engelhardt, P.V. & Beichner, R.J. Students' Understanding of Direct Current Resistive Electrical Circuits, American J. of Physics 72(1), (2004) 98-115.
3. Fleury, A.E. Programming in Java: Student-Constructed Rules, Proceedings 31st SIGCSE Tech. Symposium on Computer Science Education, ACM Press, (2000) 197- 201.
4. Gibson, I.R., Goldman, F.H. & Grinfeld, N. Mathematics at the School-University Interface in Scotland, Report 7: 2005 (Dept. Mathematics, University of Strathclyde).
5. Hirsch, L.S. & O'Donnell, A.M. Representativeness in Statistical Reasoning: Identifying and Assessing Misconceptions, Journal of Statistics Education 9(2), (2001).
6. Taber, K.S. Chemistry Lessons for Universities? A Review of Constructivist Ideas, University Chemistry Education 4(2), (2000) 63-72.
7. Zeilik, M., Schau, C. & Mattern, N. Conceptual Astronomy: Replicating Conceptual Gains, Probing Attitude Changes Across 3 Semesters, American J. of Physics 67(10), (1999) 923-927.
8. Hamza, K.M.: Describing and Analyzing Learning in Action: An Empirical Study of the Importance of Misconceptions in Learning Science, Journal of Science Education 92, 141-164, (2008).
9. Muller, D.A., Bewes, J., Reimann, P.: Saying the Wrong Thing: Improving Learning With Multimedia by Including Misconceptions, J. of Computer Assisted Learning, 24, 144-155, (2007).
10. Bull, S., Mabbott, A., Gardner, P., Jackson, T., Lancaster, M., Quigley, S.: Supporting Interaction Preferences and Recognition of Misconceptions with Independent Open Learner Models, AH2008, Springer-Verlag, Berlin Heidelberg, 62-72 (2008).
11. Bull, S., Kay, J.: Workshop on Metacognition and Open Learner Models, Metacognition and Self-Regulated Learning in Educational Technologies. Int. Tutoring Systems (2008)
12. Bull, S., Britland, M. Group Interaction Prompted by a Simple Assessed Open Learner Model that can be Optionally Released to Peers, PING Workshop, User Modeling (2007).
13. Bull, S., Broady, E. Spontaneous Peer Tutoring from Sharing Student Models, AIED1997, IOS Press, Amsterdam, 143-150, (1997).
14. Upton, K., Kay, J. Narcissus: Group and Individual Models to Support Small Group Work, UMAP2009, 54-65, (2009)
15. Dufresne, A., Hudon, M. Modeling Learner Preferences for Embodied Agents: Experimenting with the Control of Humor, Workshop on Individual and Group Modelling Methods that help Learners Understand Themselves, ITS'02, 43-51, (2002).
16. Chi, M.T.H. Commonsense Conceptions of Emergent Processes: Why Some Misconceptions are Robust, Journal of the Learning Sciences 14(2), 161-199, (2005).

Modeling Successful versus Unsuccessful Threaded Discussions

Jeon-Hyung Kang, Jihie Kim and Erin Shaw

University of Southern California Information Sciences Institute
4676 Admiralty Way, Marina del Rey, CA, U.S.A
{jeonhyuk, jihie, shaw}@isi.edu

Abstract In this paper, we present a new approach for modeling student discussion threads using a state transition model. The goal is to understand patterns of student interactions that lead to collaborative learning and result in better performance in the class. The context is a question and answer type course discussion board that is used by undergraduate computer science students. A four-state model was developed based on an analysis of sample discussion threads. Different patterns of the ending state are compared for different group interactions such as interactions with and without instructor.

Keywords: Student online discussions, discussion state model

1 Introduction

Online discussion boards are used to support collaboration in distance education and web enhanced courses. For project-based programming courses, collaboration can be critical for overcoming obstacles to timely project completion. In this work, we examine student discussions from a project-based undergraduate computer science course at the University of Southern California. Although the course is taught the same way every semester, student use of the class discussion board varies across semesters. Poor student participation, the absence of instructor participation, technical difficulties, and language barriers can all affect the success of the medium.

Our goal is to be able to discern the various causes of success/failure by studying student interactions. In this paper we examine whether it is possible to characterize successful versus unsuccessful question and answer (Q&A) type discussions. We present a state transition model for analyzing discussion threads and show how the model can be used for evaluating and comparing student discussions from different semesters. Our dataset consisted of forty randomly selected discussion threads -- twenty long threads (length greater than six messages) and twenty short threads (length less than six messages) -- out of a corpus of 177 threads.

2 Characterizing Successful vs. Unsuccessful Threads with a State Transition Model

In a Q&A type discussion, if an information seeker's problems get resolved, we can say that the discussion reached a successful conclusion, or simply, that the discussion was successful. What, then, makes one discussion thread successful and another unsuccessful? Successful discussions can be defined in many ways: "Is the initial problem resolved?" "How many answers do we have?" "Is instructor involved in this discussion thread?" and so on. We define *successful discussion* as a discussion in which all of an information seeker's questions get resolved, including initial questions, related questions, similar questions, and questions about derived problems. A four-state model was developed based on an analysis of sample discussion threads: An *initiation* state, an *understanding* state, a *solving* state and a *closing* state.

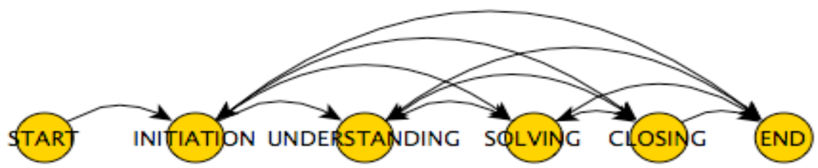


Fig. 1. State Transition Model

In the first state (initiation), there must be a problem that exists, which is almost always proposed by the information seeker. A problem is an issue that makes it difficult to achieve a desired goal, and its context, which may include the author's unique situation, issue or proposition. In the second state (understanding), the problem is elaborated through communication with other users, who need to understand why this problem exists, and need to identify the obstacles that the user confronts. Multiple messages may be required in this state, depending upon the problem type, for example, whether the problem is common or unusual, and the author's ability to clearly describe the problem. In third state (solving), information providers give solutions, instructions, propositions, or hints that suggest solutions or actually solve the problem. The solving state may also require multiple messages if the information does not satisfy the seeker. Or, for example, if the information provider misunderstands the root cause of the problem, the state machine might transition back to the understanding state.

In Figure 2, we describe four discussion thread examples with the transition model Threads **a.** and **c.** are long, and threads **b.** and **d.** are short. We labeled user roles (seeker or provider), message roles (sink or source), and speech acts, such as question, instruction, description done, issue, and proposition that can be automatically labeled by our classifiers [1,2]. Thread **a.** has all four states in sequence, ending with a closing. Thread **b.** doesn't go through the understanding state and closing is missing, but it ends with a solving state without an additional issue. Threads **c.** and **d.** are both considered unsuccessful since thread c. ends at the seeker's initiation state and thread d. ends at the provider's understanding state. There are other possible patterns that can be captured with the transition model. For example, for difficult problems, students may stay in the understanding or solving state longer than average, making the thread longer. The model also provides clues about the type of student participation such as *problem initiator*, *problem elaborator* (*in the understanding state)*, *problem solver*, etc. We expect that the state transition model will help us characterize the issues and qualitatively profile students.

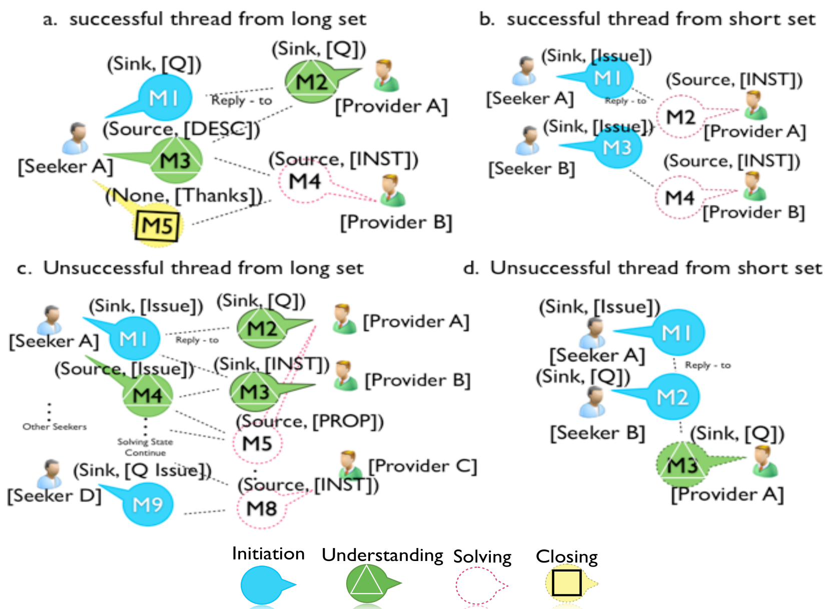


Fig. 2. Discussion thread examples (**a**: I-U-S-C | **b**: I-S-I-S | **c**: I-U-S-I | **d**: I-U)

Table 1. State transition matrix examples.

State	Initiation	Understanding	Solving	Closing
Initiation	**I:** 1. Other seekers' agreement on initial seeker's problem, e.g., "Me too, I 'm getting the same error."	**U:** 1) To understand seeker's problem, e.g., "How did you propagating your dirty bit?"	**S:** 1) Provider's answer with instruction, disagree, agree, proposition, or question, e.g., "Try putting them in a loop." 2) Seeker found his answer by himself, e.g., "I just got this."	
U	Same as **I.**	Same as **U.**	Same as **S.**	
Solving	1) Same as **I**. 2) Same Seeker reports derived problems, e.g., "I found the reason of the problem and now I have another problem…"	1) Seeker explains details about his/her problem and Providers ask about problem detail again until they understand the seeker's problem correctly, e.g., "Then where did you set the flag?"	1) Same as **S.** 2) Seeker does not understand Provider's answer message, e.g., "Where exactly can I get semaphore?"	**C**: 1) Seeker thanks the Provider 2) Provider gives praise to Seeker for solving the problem.
Closing	1) Same as **I**. 2) Same Seeker reports derived problems, e.g., "I found the reason of the problem and now I have another problem…"		1) If closing is reached by non-initial problem provider and if initial problem was not resolved, he can bring state into Solving state, e.g., "But I am still getting bus error from …"	Same as **C**.

Table 2. State transition matrix frequencies.

	Total (N)		Initiation %		Understanding %		Solving %		Closing %	
State	L	S	L	S	L	S	L	S	L	S
I	36	20	11	5	11	15	**77**	**80**	-	-
U	11	-	-	-	**81**	-	18	-	-	-
S	94	11	12	27	-	-	**82**	**54**	4	18
C	3	-	66	-	-	-	-	-	33	-

Table 1 shows the circumstances for which state transitions occur and Table 2 shows their frequencies, with significant frequencies highlighted. Each row represents a current state and each column represents the next state. For example, from understanding state to understanding transitions occur 9/11 times in the long set, and 1/1 time in the short set. We found that there are a few cases where students go through the 'understanding' state due to lack of clarity; but in most cases discussions go directly to the 'solving' state without discussing the presented issue. The solving state often consists of several messages before it reaches the final message or the closing state but shorter threads tend to contain a short solving state.

Table 3 shows the results of analyzing transitions in discussions between instructors and students. We found that the instructor's role in this course is very important, when we compare the average lengths of the discussion threads. Furthermore, when the instructor participated in the thread, the discussion ended at the solving or closing state in 26 out of 31 threads.

Table 3. Types of threads and their ending states.

Author	Average thread length	Number of threads ending at this state:			
		Initiation	Understanding	Solving	Closing
Instructor	6 msgs	3	5	23	3
Students only	2.2 msgs	3	0	3	0

3 Discussion

It is difficult to understand the many different types of student interactions that occur in online discussions, and even more difficult to understand how they affect collaborative learning. To assess successful versus unsuccessful interactions, we will compare state transition patterns from different semesters and relate them with student performance. By using large a data sample, we hope to better understand patterns of successful collaborative discussions while considering other factors such as user response time and degree of participation, and to analyze them further.

4. References

[1] Kang, J. and Kim, J. 2010. Profiling Message Roles in Threaded Discussions using an Influence Network Model, internal project report.

[2] Kim, J., Kim, T. and Li, J., 2009. Identifying Unresolved Issues in Online Student Discussions: A Multi-Phase Dialogue Classification Approach, Proceedings of the AI in Education Conference, 2009.

Conversational Tutors with Rich Interactive Behaviors that support Collaborative Learning

Rohit Kumar, Carolyn P. Rosé

Language Technologies Institute, Carnegie Mellon University,
5000 Forbes Avenue, Pittsburgh, Pennsylvania 15213
{ rohitk , cprose } @ cs.cmu.edu

Abstract. In addition to presenting relevant instructional content to learners in collaborative learning settings, intelligent conversational tutors must be good communicators to keep the students attentive and engaged. In our recent work, we have shown that tutors capable of performing both task-related (tutoring) as well as social behavior are more effective tutors than tutors that perform only task-related behavior. It is also important to consider that collaborative learning tasks in ill-defined domains (like engineering design) require students to negotiate among competing goals. Thus, automated tutors must be capable of supporting students with different roles and potentially competing goals. In this paper, we will describe the most recent implementation of the CycleTalk tutor using the Basilica architecture that demonstrates these social behavior and role supporting capabilities. Also, we briefly discuss results of an experiment using this tutor that optimizes the amount of social behavior in such learning settings.

Keywords: social interaction, conversational agents, collaborative learning

1 Introduction

Conversational Tutors used in state of the art tutorial dialog systems have been shown to be effective support for collaborative learning groups [1]. Investigations on this promising educational technology have focused on exploring tutoring behaviors that are effective in delivering instructional content (like feedback, hints, interactive dialogs, etc.) relevant for the learning task in a timely manner.

In this paper, we will describe a recent implementation of the CycleTalk tutor, which we have used as a vehicle for delivering and evaluating such interactive behavior. CycleTalk is an intelligent tutoring system that helps sophomore engineering students learn principles of thermodynamic cycles (specifically Rankine Cycle) in the context of a power plant design task. Teams of two students work on designing a Rankine cycle using a Thermodynamics simulation software package called CyclePad [2]. As part of the design lab during which this learning task is performed, students participate in collaborative design interaction for 30-45 minutes using ConcertChat, a text based collaboration environment [3]. The CycleTalk tutor participates in the design interaction along with the two students to provide instructional support and help the students learn underlying thermodynamic concepts as they design.

One of the interactive conversational behaviors we have recently studied is motivated from research in the field of small group communication. Empirical studies in this field [4] have shown that human participants in such groups perform both task-related as well as socio-emotional interactive behaviors. Bales' Interaction Process Analysis (IPA) schema identifies six categories that are related to positive as well as negative socio-emotional interactive behaviors. Using tutors capable of performing these social behaviors to support students in a freshmen mechanical engineering collaborative design task, we have shown that such tutors are significantly better than tutors that have no social behavioral capabilities [5]. In this paper, we will describe the implementation of eight social interaction strategies (Table 1) corresponding to two of the positive socio-emotional categories identified by IPA.

Table 1. Social Interaction Strategies

1. Showing Solidarity: *Raises other's status, gives help, reward*
1a. Do Introductions: *Introduce and ask names of all participants*
1b. Be Protective & Nurturing: *Discourage teasing*
1c. Give Re-assurance: *When student is discontent, asking for help*
1d. Complement / Praise: *To acknowledge student contributions*
1e. Encourage: *When group or members are inactive*
1f. Conclude Socially
2. Agreeing: *Shows passive acceptance, understands, concurs, complies*
2a. Show attention: *To student ideas as encouragement*
2b. Show comprehension / approval: *To student opinions and orientations*

The CycleTalk power plant design task involves designing a thermodynamic cycle that has multiple desirable design goals. Within practical constraints, these design goals conflict with each other. In order to help the students consider all of these goals in their designs, students participating in the design task are assigned one of two roles (pro-environment and pro-power production) that favor different design goals. In the next section, we will discuss implementation of behaviors that the CycleTalk tutor exhibits as a participant of the design group to identify student goals and providing instructional content to support specific goals or remain neutral.

2 Implementation of the CycleTalk tutor

The CycleTalk tutor has been implemented using the Basilica architecture [6] which provides the flexibility and representational power to build conversational agents that can exhibit rich interactive behaviors like the ones mentioned in the previous section. Agents built using the Basilica architecture are composed of a network of behavioral components that communicate and coordinate among each among through events propagated over the component network.

The component network (20 components and 40 connections) of the CycleTalk tutor agent is shown in Figure 1. The *ConcertChatListener* and *ConcertChatActor* components provide connectivity to the ConcertChat environment and isolate the components of the agent to allow easy integration with other environments if required. Text messages from the students are propagated through the component network after being annotated with semantic categories by the *AnnotationFilter*.

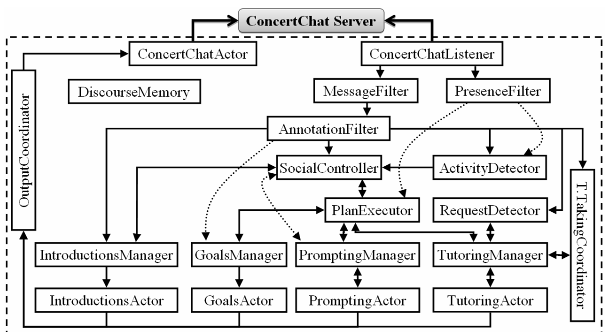


Fig. 1. Component Network of the CycleTalk Tutor

The interactive behaviors of the agent that are directly observable by the students are implemented by 4 manager-actor dyads. For example, the *IntroductionsManager-IntroductionsActor* dyad implements the introductions behavior that is performed when the Social Interaction Strategy 1a is triggered. The *PlanExecutor* and the *SocialController* trigger relevant task-related and social behavior respectively.

The *PlanExecutor* executes the tutor's task-related interaction plan comprised of 14 steps (including 4 tutorial dialogs) some of which may be skipped in the interest of time. One of the steps includes asking the students about their design goals which is performed by the *GoalsManager-GoalsActor* dyad. Based on a configuration parameter, the *TutoringManager* can favor one of the goals (or remain neutral) by choosing corresponding versions of the 4 tutorial dialogs when they are triggered.

The *SocialController* implements the eight social strategies listed in Table 1. The strategies are triggered by rules based on the most recent plan step (for strategy 1a, 1d, 1f), semantic categories of the most recent student turns (for strategies 1b, 1c, 3a, 3b) and inactivity events by the *ActivityDetector* (for strategy 1e). In additional to these rules, the amount of social behavior is regulated using a Social Ratio parameter that specifies the percentage of all tutor turns that can be generated by the *SocialController*. For example, Social Ratio set at 20% limits the tutor to perform at most 20 turns generated by the *SocialController* for every 100 turns by the tutor.

3 Choosing optimal amount of Social behavior

We conducted an experiment to evaluate the effect of amount of social behavior performed by the CycleTalk tutor. 106 sophomore Mechanical engineering students enrolled in a sophomore course participated in the experiment. The students worked in teams of two to design a power plant. Teams were randomly assigned to one of three conditions corresponding to different amounts of social behavior exhibited by the tutors. The three conditions included 1. No Social Behavior (*None*), 2. Tutor with Social Ratio of 15% (*Low*) and 3. Tutor with Social Ratio of 30% (*High*).

Pre-test and post-tests were administered to the students individually before and after the design interaction with the tutor. The pre-test had one additional question than the post-test which was added to make the pre-test and post-test slightly different. 22 objective (multiple-choice) and 6 subjective (brief-essay) type questions common to both the tests were used to compute learning outcomes (Table 2).

Table 2. Average Pre & Post test scores for each condition (Standard deviation in paranthesis)

Condition	Pre-Test			Post-Test		
	Total	Objective	Subjective	Total	Objective	Subjective
None	13.94	11.28	2.67	17.72	13.33	4.39
0% Social	*(4.53)*	*(2.91)*	*(2.23)*	*(4.09)*	*(2.47)*	*(2.04)*
Low	14.00	11.38	2.62	18.59	14.77	3.82
15% Social	*(6.15)*	*(4.16)*	*(2.54)*	*(4.72)*	*(3.43)*	*(1.74)*
High	14.08	12.03	2.06	17.72	13.75	3.97
30% Social	*(4.46)*	*(3.13)*	*(1.88)*	*(3.77)*	*(3.07)*	*(1.72)*

ANCOVA models for the three types of scores using pre-test scores as a covariate and condition as independent variables showed that there were no significant differences between the three conditions on the total and the subjective scores. However, there was a significant effect of the condition variable on the objective scores $F(2, 97)=3.48$, $p < 0.05$. A pairwise Tukey test post-hoc analysis showed that the *Low* (15%) condition was marginally ($p < 0.07$) better than both *None* (effect size = 0.69σ) and *High* (effect size = 0.55σ) conditions. Besides confirming prior results [5], this study shows that excessive display of social behavior can be detrimental to task performance (learning). It is important to design tutors that perform an appropriate moderate amount of social behavior in collaborative learning settings.

Acknowledgments. The research was supported by NSF grant number DUE 0837661

References

1. Kumar, R., Rosé, C. P., Wang, Y. C., Joshi, M., Robinson, A.: Tutorial Dialogue as Adaptive Collaborative Learning Support. In: Proc. of AI in Education (2007)
2. Forbus, K. D., Whalley, P. B., Evrett, J. O., Ureel, L., Brokowski, M., Baher, J., Kuehne, S. E.: CyclePad: An Articulate Virtual Laboratory for Engineering Thermodynamics. Artificial Intelligence. vol. 114 (1-2), pp. 297-347 (1999)
3. ConcertChat, http://www.ipsi.fraunhofer.de/concert/index_en.shtml?projects/chat
4. Bales, R.F.: Interaction process analysis: A method for the study of small groups, Addison-Wesley, Cambridge, MA (1950)
5. Kumar, R., Ai, H., Beuth, J. L., Rosé, C. P.: Socially-capable Conversational Tutors can be Effective in Collaborative-Learning situations. To Appear In: Intl. Conf. on Intelligent Tutoring Systems, Pittsburgh, PA (2010)
6. Kumar, R., Rosé, C. P.: Basilica: An architecture for building conversational agents. In: Proc. of NAACL-HLT, Boulder, CO (2009)

Agents for Collaborative Learning in Virtual Worlds

Chunyan Miao[1], Han Yu[1], Zhiqi Shen[2],
Xuehong Tao[3]

[1] School of Computer Engineering, Nanyang Technological University, Singapore
[2] School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore
[3] School of Commmunication and Information, Nanyang Technological University, Singapore
{ascymiao, yuha0008, zqshen}@ntu.edu.sg, xuehongtao@gmail.com

Abstract. Virtual worlds provide educators with rich environments to explore the application of advanced learning theories in collaborative learning which is hard to achieve in traditional classroom based settings. However, the process of converting these theories into useful scaffoldings in such unique environments is highly challenging. In this paper, we explore the use of intelligent software agents to infuse advanced learning theories into the interactions among the learners as well as the learning environment. We propose a collaborative teachable agent which incorporates learning-by-teaching-others theory into a virtual learning environment (VLE) we have developed. We aim to set up conditions conducive for collaborative learning in VLEs using this agent. The effects of the proposed agent on the learning experience of the students have been verified in a field study in one of the high schools in Singapore.

Keywords: intelligent software agents, teachable agents, virtual learning environment, collaborative learning.

1 Introduction

Recent advances in computer graphics technologies enabled virtual worlds to achieve levels of immersion not seen before through a heightened sense of presence and the suspension of disbelief. High quality, low bandwidth graphics rendering makes accommodating massive numbers of users in a virtual world possible. These two key developments have catapulted online virtual worlds to a high level of fame especially among the "Net Generation" [4]. The virtual worlds, which offer users a connected computer simulated environment where people can interact via their avatars, provide educators with a powerful medium to experiment with new ideas in collaborative learning that can reach out to a large audience [1]. The use of software agents to incorporate learning activities into virtual learning environments (VLEs) has been advocated by pioneer researchers [2], [3].

This research is supported by the Singapore Millennium Foundation (SMF) and the National Research Foundation (NRF), Singapore.

Augmenting VLEs with software agents has the potential to open up vast opportunities not only for providing infrastructural support for collaborative learning (e.g. communications, learning task management and personalization, etc.), but also for engaging the students in a deeper cognitive level to facilitate the construction of useful scaffoldings through the application of the learning-by-teaching-others theory during the collaborative learning process. In this paper, we explore how collaborative teachable agents (TAs) can be designed to assist knowledge creation within a group of learners.

2 Related Work

Learning by teaching others is a well known educational theory [5]. When in a position to prepare to instruct others instead of just learning for one's own knowledge gain, the students often take on a higher sense of responsibility. This motivates them to organize the knowledge they have acquired in anticipation of the needs from their pupils. In the field of teachable agent research, the most popular one so far is the agent Betty [6]. It is an agent using a cognitive map (CM) to represent its knowledge. By doing so, Betty is able to understand and infer the causal relationships among various concepts. Nevertheless, due to the limitation of CM being able to denote only the causality but not the degree of causality between any two concepts, the level of sophistication of the concepts that users can teach Betty is limited. Moreover, the Betty family of teachable agents has not made provision for collaborative learning situations.

3 The Collaborative Teachable Agent

Fig. 1. The Rule Creator Interface for the TA

The collaborative TA is designed to support the learning-by-teaching-others theory in small groups of collaborating learners. The main objective is to infuse intelligent behaviors into the TA We incorporated inquiry based learning and collaborative learning in the TA based on fuzzy reasoning in an agent controlled avatar – "Little Banana Tree" – in CS. It enables the students to reflect upon what they have learnt in the VLE and to counter check the coherence of their knowledge. After the learning tasks are completed, the students can use the rule creator interface as shown in Fig 1 to collaboratively incorporate knowledge to the TA in the form of rules (i.e. if…then…else… clauses). The learners are free to decide the level of details of the rules to facilitate the checking process for any rule conflict. Once new rules are entered, the TA checks them against the existing rules on the related concepts in the rule database for potential conflicts. If a conflict is found, the TA will generate alerts of conflicting rules to the group of collaborating students to prompt them to resolve the conflict. Through this mechanism, discussions and consensus building among the group members will be stimulated and they will become aware of their potential misunderstandings thereby, improve their knowledge on the subject topic.

The TA constantly monitors the learning activities the students are engaged in. When it recognizes a known concept has been involved in the current activity, it synthesizes an enquiry for the students to answer. The enquiry is produced making an inference with the recognized concept. For example, "does more sunshine lead to higher temperature?" An error bias is added into the TA's configuration which can introduce a degree of random error into the inference result to make the inference result incorrect. This may prompt the learners to think about the problem from a different angle and correct the TA's mistake. Through this interaction, the students could reflect deeper on the relationship of the various knowledge points they have learnt and possibly form a more in-depth understanding of the various concepts.

4 Field Study

A prototype VLE – the Chronicles of Singapura – equipped with the collaborative TA has been deployed in a secondary school in Singapore for science class topics. A total of 68 secondary two (i.e. Grade 8 in the U.S. system) students participated in the study. The students were divided into two groups according to their respective classes. Students formed into sub-groups of either 3 or 4 to collaboratively explore the VLE as a team. While learning concepts related to the transportation of materials in plants through osmosis, diffusion and active transport mechanisms as well as the process of photosynthesis.

A survey has been conducted to evaluate the students' learning experience with the TA. Some of the comments in the questionnaire from the students include: "*I could do things like fly, float and walk on water, I also could shrink and teleport. This is not life-like, and it is fun, as it receives a lot of imagination*"; "*I found the process of progressing up the xylem and answering questions rather innovative and enjoyable. We actually had to rise up the xylem, which demanded a lot on scientific information, we recalled from Uncle Ben's teaching*"; "*Working together with my teammates to achieve much more, for instance, we would split ourselves into smaller groups, one* *concentrating on creating food while the other two collect and transport it*". The overall rating for the learning experience was 5/7.

The TA offers the students a satisfactory learning experience. The novel learning activities that allow students to participate in experiences that are not possible in the real world have elicited positive interest in the VLE from the students. The TA is perceived as helpful by the groups of collaborating students.

5 Conclusions and Future Work

In this paper, we have explored the opportunity to assist learning scaffolding construction in collaborative learning with learning activities revolving around TAs. We have conducted a preliminary study of the prototype system in a secondary school in Singapore. The preliminary assessment on user acceptance and feedback on the effectiveness of the TA is encouraging. Additional field studies will be conducted in our future work to verify the effectiveness of the proposed TA more formally.

Nevertheless, there are still many research issues worth exploring for the novel use of agents to assist collaborative learning. In a socially complex learning environment such as a VLE, the emotions derived from the interactions with other people or autonomous entities, such as agents, is an important factor that can affect the learning experience and learning outcome of a student [7]. Therefore, it may be beneficial for agents in collaborative learning situations in such environments to be equipped with affective capabilities. These include incorporating emotional response to students' actions, inferring students' emotional states at runtime and eliciting desired emotional states from students by means of simulated emotions or verbal or visual stimuli. These issues will be further investigated in our subsequent studies.

References

1. Bainbridge, W. S.: The Scientific Research Potential of Virtual Worlds. Science, 317, 472 – 476 (2007).
2. Dede, C.: Immersive Interfaces for Engagement and Learning. Science, 323(5910), 66-69 (2009).
3. Jacobson, M. J., Miao, C., Kim, B., Shen, Z., & Chavez, M.: Research into Learning in an Intelligent Agent Augmented Multi-user Virtual Environment. In: IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, pp. 348-351 (2008).
4. Oblinger, D. G. and Oblinger, J. L.: Educating the Net Generation, Edu-cause e-book, pp. 227-236 (2005).
5. Cohen, P. A., Kulik, J. A., & Kulik, C.-L. C.: Educational Outcomes of Peer Tutoring: A Meta-analysis of Findings. American Educational Research Journal, 19, 237-248 (1982).
6. Blair, K., Schwartz, D. L., Biswas, G., & Leelawong, K.: Pedagogical Agents for Learning by Teaching: Teachable Agents. Educational Technology & Society (Special Issue on Pedagogical Agents) (2006).
7. Immordino-Yang, M.H. and Damasio, A.: We Feel, Therefore We Learn: The Relevance of Affective and Social Neuroscience to Education. Mind, Brain and Education, 1, 3-10 (2007).

Assessing, Modeling, and Supporting Helping Behaviors in Computer-Mediated Peer Tutoring

Erin Walker[1], Nikol Rummel[2], Kenneth R. Koedinger[1]

[1]Carnegie Mellon University, Pittsburgh, PA, USA
[2]Ruhr-Universität Bochum, Bochum, Germany
erinwalk@andrew.cmu.edu, rummel@psychologie.uni-freiburg.de,
koedinger@cmu.edu

Abstract. Adaptive collaborative learning support is a promising new method for facilitating student collaboration, in that support would be tailored to individual student needs. To investigate, we built the Adaptive Peer Tutoring Assistant (APTA) to support peer tutoring interactions in high school mathematics. The system uses existing problem-solving models from the Cognitive Tutor Algebra, a successful individual intelligent tutoring system, as a basis for interaction models that provide assistance to peer tutors on giving more conceptual help. As part of APTA, we built a model of ideal peer tutoring, a method for assessing the quality of actual peer tutoring, and several support mechanisms to guide students in tutoring more effectively.

Keywords: intelligent tutoring, computer-supported collaborative learning, adaptive collaborative learning systems, peer tutoring

1 Introduction

A promising new method for facilitating student collaboration is by providing students with adaptive collaborative learning support (ACLS), where student interaction is analyzed as it occurs by comparing it to an ideal model, and support is provided based on this comparison [1]. Studies comparing adaptive to fixed support for collaboration have indeed been promising [2], but research into ACLS is still at an early stage. In our project, we investigate how to build an effective ACLS system for improving helping behaviors during peer tutoring. We developed a peer tutoring environment as an addition to the literal equation solving unit of the Cognitive Tutor Algebra (CTA), a successful intelligent tutoring system for high-school math [3]. During peer tutoring, students in the same class are paired and collaborate with each other at different computers. Peer tutors are able to see the peer tutee's actions, but cannot solve the problem themselves. The peer tutor can give help and discuss the problem with tutees in a chat tool, by first labeling their utterance (e.g., as a hint) and then typing their message. Peer tutors can also give feedback, marking tutee actions right or wrong. In this paper, we describe support we have implemented within the context of this environment to improve peer tutor help. Although we have evaluated the system in a large-scale classroom study, due to space limitations, we will only discuss its implementation here.

2 The Adaptive Peer Tutoring Assistant (APTA)

2.1 Modeling & Assessing Peer Tutoring

In order to provide adaptive assistance to the peer tutor, we used previous research into how students benefit from giving and receiving help to construct a computational model of peer tutoring. Help should explain the reason behind a step using relevant concepts (*conceptual help*). It should be given at tutee errors (*timely help*), and target their misconceptions by explaining why a step is a mistake (*targeted help*) [4]. We used a simple model composed of 15 rules to represent how each peer tutor action mapped to these three skills. The model included an additional skill representing how accurately peer tutors used sentence classifiers (e.g., labeled their action as a prompt, error feedback, hint, or explanation). To assess peer tutor performance, the system used a combination of inputs. First, it used CTA domain models to see if tutees had recently made an error (and thus if they needed help). Next, it used student interface actions, including tutor use of sentence classifiers. Finally, it used Taghelper [5] to build a machine classifier trained on previous study data (kappa = .82 for the previous dataset) that could automatically determine whether students were giving help, and whether the help was conceptual. By using these inputs we determined which model rules fired, and then use Bayesian knowledge tracing [6] to maintain a running assessment of the three skills. More specifically, we relied solely on the machine classification to assess whether students gave *conceptual help*. We increased our assessment of student ability to give conceptual help any time an utterance was flagged as conceptual help. We decreased the skill assessment any time students self-classified their utterances as hints or explanations, and the machine classifier labeled the utterances as not conceptual. We used a combination of student self-classifications and problem-solving context to determine if students were giving *timely help.* For example, self-classified help after a tutee error was made incremented the skill, while help after the tutee took a correct step decremented the skill. *Targeted help* was assessed in a similar manner. After the tutee made an error, if peer tutors self-classified their utterance as a prompt or error feedback, we increased our assessment of this skill. If they self-classified their utterance as a hint or explanation, we decreased our assessment of this skill. To determine whether peer tutors were using sentence classifiers appropriately, we compared which classifier they selected to whether the system labeled their utterance as help.

2.2 Adaptive Assistance to Peer Tutoring

Based on the model assessment, we provided students with three kinds of assistance for their interaction support. Our first type of assistance, integrated hints, was used for instances where the tutor did not know how to help the tutee. In this case, the peer tutor could click on a hint button, found in the top right corner of the interface (see Figure 1), and receive a multi-level hint on both how to solve the problem and how to help his partner (bottom right of Figure 1). The hint opens with a collaborative component, and then contains the cognitive component that tutees would have received had they been using the CTA individually. The collaborative component is drawn from one of the four peer tutor skills we are trying to encourage, and is adaptively chosen based on the current problem-solving context (e.g., it varies depending on whether the tutee has most recently taken a correct or incorrect step). The peer tutor is intended to integrate the collaborative assistance on what kind of help to give with cognitive assistance for how the tutee should proceed.

There may be cases where even after examining the adaptive hints, the peer tutor is unsure how to use the hints to give the tutee feedback. We designed conceptual resources to further assist the peer tutor in constructing good help. For example, when the peer tutor clicks the "give hint" sentence classifier to prepare to compose a hint to his partner (located in the bottom left of Figure 1), he is presented with a resource (front and center in Figure 1), with content tailored to the current problem type, which provides examples of what a good hint would be within the context of this problem type. We had 4 separate sets of resources mapping to each type of sentence classifier (one for prompts, one for error feedback, one for hints, and one for explanations). As the resource presents several sample hints for the whole problem, the peer tutor has to actively process the resource in order to determine which kind of hint might apply.

Once the peer tutor has given help, the computer may give a reflective prompt in a chat window that appears simultaneously to both students and targets peer tutor skills

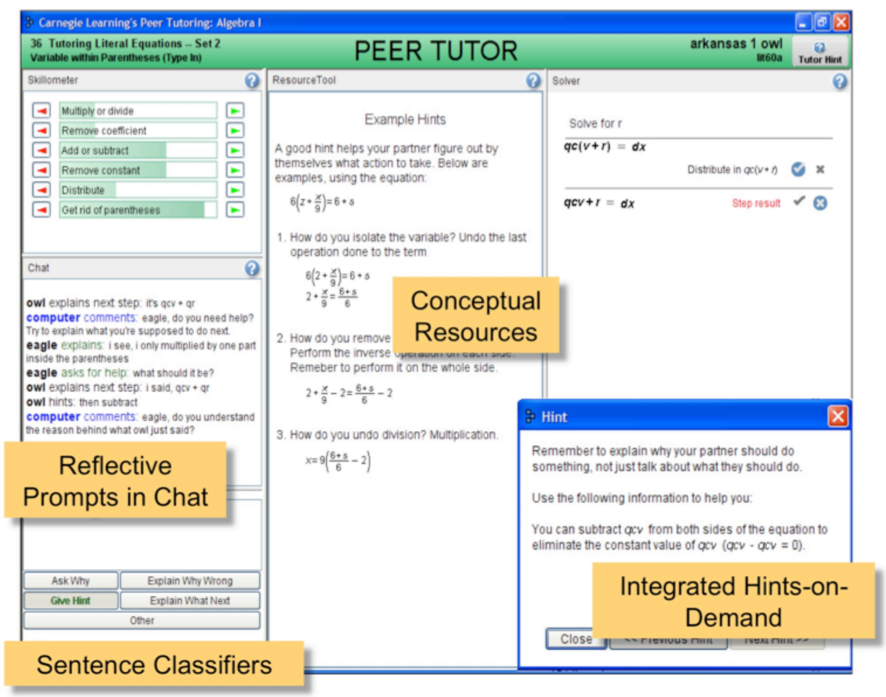


Figure 1. Assistance in APTA.

that need improvement. For example, the peer tutor may say "then subtract" rather than the more conceptual "to get rid of *qcv*, you need to perform the inverse operation." In that case, the computer uses its assessment of the peer tutor's help-giving skill to give a reflective prompt like, "Tutee, do you understand the reason behind what the tutor just said?" This utterance is designed to get both students reflecting on the domain concepts behind the next step, and to remind the peer tutor that help should explain why in addition to what. Prompts were addressed to either the peer tutor or the tutee, and were adaptively selected based on the computer assessment of peer tutor skills. Students also received encouragement when they displayed a particular collaborative skill ("Good work! Explaining what your partner did wrong can help them not make the same mistake on future problems"). Only one reflective prompt was given at a time, and parameters were tuned so that students received an average of one prompt for every three peer tutor actions. There were several different prompts for any given situation.

3 Summary

In this paper, we described a method for assessing the quality of peer tutor help, and then described how a system can support a peer tutor based on the model. This work makes contributions to the modeling of student collaboration in its application of knowledge-tracing to collaborative skills, and its assessment of help quality using a combination of domain and collaboration information. The support we have developed provides three different paths by which students can improve their collaboration quality.

Acknowledgments. This research is supported by the Pittsburgh Science of Learning Center, NSF Grant #SBE-0836012. Thanks to Carolyn Rose for her helpful suggestions.

References

1. Soller, A., Martinez, A., Jermann, P., and Mühlenbrock, M. 2005. From mirroring to guiding: A review of state of the art technology for supporting collaborative learning. IJAIED, 15, 261–290.
2. Baghaei, N., Mitrovic, T., and Irwin, W. 2007. Supporting Collaborative Learning and Problem Solving in a Constraint-based CSCL Environment for UML Class Diagrams. IJCSCL, 2 (2-3), 159-190.
3. Koedinger, K., Anderson, J., Hadley, W., & Mark, M. 1997. Intelligent tutoring goes to school in the big city. IJAIED, 8, 30-43.
4. Webb, N. M. 1989. Peer interaction and learning in small groups. I. J. of Ed. Res. 13, 21–40.
5. Rosé, C. P., Wang, Y. C. , Cui,Y., Arguello, J.,Fischer, F., Weinberger, A., Stegmann, K. (To appear). Analyzing Collaborative Learning Processes Automatically: Exploiting the Advances of Computational Linguistics in Computer-Supported Collaborative Learning. IJCSCL.
6. Corbett, A. T., & Anderson, J. R. (1995). Knowledge tracing: modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction, 4*, 253–278.

Author Index

Asterhan, Christa 1

Bader-Natal, Ari 5
Bull, Susan 9

Johan, Rasyidi 9

Kang, Jeonhyung 13
Kim, Jihie 13
Koedinger, Ken 25
Kumar, Rohit 17

Miao, Chunyan 21

Rose, Carolyn 17
Rummel, Nikol 25

Schwarz, Baruch 1
Shaw, Erin 13
Shen, Zhiqi 21

Tao, Xuehong 21

Walker, Erin 25

Yu, Han 21