Identifying the Support Needed in Computer-*Supported* Collaborative Learning Systems

Sandra Katz

Learning Research and Development Center
University of Pittsburgh

Abstract

Computer-supported collaborative learning (CSCL) systems hold the potential to enhance the effectiveness of peer learning interactions, assuming that these systems are truly *supportive*—i.e., capable of coaching collaborating peers as they work on problems and critique other students' solutions. We argue that in order to achieve this goal, CSCL system developers need to know more about the types of coaching that students are typically able to provide to each other during problem-solving activities, and what types of advice they need from more experienced students or "mentors" at various stages of their development in the instructional domain. CSCL system developers also need to know how *human* mentors provide such coaching, and—based on this information—to develop computer models of human guidance during collaborative interactions. We describe a pilot study (in progress) whose goals are to address these issues and to assess the effectiveness of the research methodology we devised for doing so.

Keywords — Technological developments for CSCL, problem solving, theories of collaboration and learning.

1. Introduction

Collaborative use of instructional systems can take on a variety of forms—from two or more students working on problems at the same computer workstation, using a tutoring system that was primarily designed with an individual user in mind, to peer collaboration on systems which were specially tailored for use by multiple learners working at the same workstation or across networked machines. The latter are typically called *computer-supported collaborative learning (CSCL)* systems, since they are intended to "scaffold" or *support* students in working together productively. Various types of support can be provided by these systems, including utilities for communicating ideas and information, facilities to access documents and other types of information, advice during problem-solving activities, etc.. The type of support that we focus on in this paper is advice during problem solving.

But why scaffold collaborative problem solving? Despite an abundance of studies showing that social learning situations correlate with a wide range of positive outcomes—including greater learning, increased productivity, more time on task, transfer of knowledge to related tasks, higher motivation, and heightened sense of competence (e.g., Slavin, 1990)—there is also widely recognized room for improvement. Collaborative learning does not work for all learners, and the results of instructional outcome studies are mixed (Webb, 1987). Fruitful student interactions are simply not a given. In Brown and Palincsar's (1989) words:

> *Social interactions do not always create new learning; peer interactions vary enormously; only some teaching environments actually create ideal learning experiences.* (p. 397)

Thus, one of the prime factors motivating the development of CSCL systems is to improve the effectiveness of collaborative learning as an instructional format (e.g., McManus & Aiken, 1993); to "create ideal learning experiences," as Brown and Palincsar invite instructional designers to achieve.

In recent years, several researchers have attempted to find out what makes some peer interactions successful but not others. One important line of research has focused on verbal exchanges during group work (e.g., King, 1989; Webb 1987, 1989). This work converges on an important finding: i.e., that the *nature of peer interactions* is perhaps the most critical factor mediating individual student achievement. As Webb and her colleagues have consistently shown through over a decade of research on peer interactions in computer-based and non-computer-based settings, "students learn more by giving elaborated help to others and learn less by receiving low-level elaboration from others" (Webb & Farrivar, 1993). However, unelaborated explanations or "terminal responses" are more typical of student responses than are such "high quality explanations"—at least among school-aged children (e.g., Webb, 1989). Not surprisingly, research by King (1989) shows that another important interaction skill which students often lack is the ability to ask questions which evoke elaborated explanations.

The results of this research on the relation between the nature of peer interactions and learning are important for developers of CSCL systems, because they suggest that these systems should scaffold the production of high-level questions and elaborated explanations. However, the research done to date is weak in accounting for when and how—i.e., in what *contexts*—peer question-asking and explanation failures occur. Various issues need to be addressed in order to tailor advice in CSCL systems according to the advising capabilities of the target users, such as: Are students uniformly and consistently poor at giving explanations, or do they tend to explain some types of knowledge more readily than other types—e.g., procedural knowledge as opposed to conceptual or strategic/planning knowledge? To what extent are explanation failures correlated with (and perhaps indicative of) knowledge gaps? As knowledge and skill in the task domain increases, does the quality of explanations (and questions) improve? How do collaborating peers use the various information and advising resources available to them in a learning environment? How do human mentors scaffold collaborating students during impasses—e.g., when is mentors' support directive; when is it more dialectical? With these research issues in mind, we observed and videotaped students as they worked together on **Sherlock II** (e.g., Lesgold et al., 1992), a computer-based, coached practice environment developed to train U.S. Air Force avionics technicians to diagnose electronic faults in F15 aircraft modules and in the complex testing systems used to check out these modules. Our principle aims in this pilot study are to: (1) capture the nature of peer learning interactions in this problem-solving domain, in terms of the issues raised above, (2) determine the extent to which the findings derived from this analysis could inform the development of effective advising capabilities in a collaborative version of **Sherlock II**, and (3) assess the effectiveness of our research methodology for accomplishing the preceding objectives. The data collection phase of the study is completed, and we have started to analyze the data. In this paper, we describe the pilot study and the data that it has yielded.

2. Description of the Pilot Study

2.1. Research Context

Like its predecessor, **Sherlock**, **Sherlock II** is a coached practice environment. It provides a realistic computer simulation of the actual job environment, allowing students to make measurements, interpret readings, replace suspect components with shop standards, etc., until they have isolated the faulty component. Students thus acquire and practice skills in a context similar to the real context in which they will be used. Advice is available "on demand" throughout problem solving. After solving a problem, students enter a review phase which we call Reflective Follow-up (RFU). During RFU, students can step through their solution with feedback from the computer coach, ask to see a sample expert solution, get advice for their next session, and review the standards of effective troubleshooting. In other words, RFU "debriefs" students on their performance.

2.2. Subjects

Three groups of people participated in this pilot study of peer interaction in a coached practice environment. First, eight experienced avionics technicians from local (Pittsburgh-based) Air National Guard and Air Force Reserve units served as *mentors*. Among these mentors, on-the-job experience in avionics ranged from three to twenty years (7.5 years was the mean). The second group of participants consisted of 16 *avionics students* (forming eight dyads) from two local avionics technical schools. All but three avionics students were in their final term of a six-term program; the other three were in their fifth term. Finally, four *Air Force ROTC students* (forming two dyads) also served as subjects. All eight mentors, and all but two of the twenty students, were male. Subjects volunteered to participate, and were payed a nominal amount.

2.3. Procedure

Following an orientation to the **Sherlock II** task domain and practice with using the tutor, all but two of the ten dyads were assigned to a mentor, who worked with the same dyad for one two-hour session per week. All dyads worked on the same set of **Sherlock II** problems, in the same order. There were two phases of data collection. In the first phase, students worked together at the same computer, doing one or two problems per session. One student was assigned the role of problem solver; the other student was assigned the role of "coach." These roles were reversed for each successive problem, in *reciprocal teaching* fashion (Palincsar & Brown, 1984). Students were directed to ask their peer for advice first, to use Sherlock (the computer coach) if their peer could not answer their question (or answer it adequately), and to ask the human mentor if they were still stuck or had a question about something in Sherlock's message. In this way, requests for "external" (i.e. non-peer-provided) coaching will be clearly marked in the data, allowing us to readily identify instances of peer coaching failures.

The human mentor observed the students' actions from a networked machine. Although the two machines were in the same room, students were directed not to talk to their mentor. Instead, students were asked to pretend that they were avionics technicians stationed in Alaska, while their mentor was in Hawaii; their only form of communication was via a teletyping window on their and their mentor's screen. Since one of our main goals is to capture how mentors scaffold students when the latter are unable to coach each other during collaborative work, we placed one restriction on mentors' behavior: we asked them to let students try to help each other first, and to initiate advice only when the mentors thought that doing so would prevent students from "going down the garden path." We also told mentors that one way they could respond to students' queries was by suggesting that students access one of Sherlock's coaching options.

As illustrated in Figure 1, the teletyped communication between students and their human mentor was automatically recorded and integrated with the transcript of student actions that the system generates during problem-solving sessions. These transcripts will allow us to analyze students' interactions with their human mentor in the context of students' problem-solving actions.

The first phase of data collection, combining videotaped with teletyped recording of dyad-mentor dialogues, lasted for two sessions per dyad. During the second phase of data collection, students worked at *separate* machines. As before, one student acted as problem solver, the other student as "coach." However, communication was entirely through the teletyping mechanism described above, which students used during phase one to communicate with the human mentor. During this data collection phase, the human mentor's role was explicitly to "coach the student coach." The mentor and student coach did this via teletype, using a second keyboard that was connected to the student coach's machine. When the student coach was unable to respond to his peer's request for help, he could carry out a teletyped dialogue with the mentor until he reached the point where he could advise his peer. When this happened, he transmitted the dialogue he had with his mentor to the student solving the problem. In order to be able to distinguish in the transcripts between the dialogue contributions of the mentor and the student coach, the mentor typed in capital letters; the student coach typed in regular font. This second phase of data collection consisted of four sessions per dyad.

We separated data collection into these phases for two reasons. First, we want to be able to compare videotaping with teletyped communication for studying dyadic interaction, and human mentors' support for the same, in computer-based learning environments. In particular, to what extent, and in what ways, does teletyping reduce the communication bandwidth? Does the convenience and contextualization provided by the automated transcripts compensate for the reduced bandwidth of teletyped communication? Second, several of the students and a few of the mentors were poor typists. So, in order to prevent "cognitive overload," we tried to minimize the amount of typing that they would have to do during early sessions, when they were still getting comfortable with the **Sherlock II** task domain and user interface. By the third session, students were adept enough **Sherlock II** users to communicate with each other via teletype, although it was somewhat "slow-going" for the poorest typists.

3. Conclusion

Several researchers have been developing instructional software which is specially tailored for collaborative use —i.e., CSCL systems. However, very few CSCL systems are being built in such a way that the advising resources available to students are based on empirical research on what students actually do, and fail to do, while they collaborate. [Baker and Bielaczyc's (1995) research in identifying "missed opportunities" (MOs) for conceptual development when students work together in the context of the CHENE CSCL system is a notable exception.] We believe that CSCL environments hold the potential to enhance the effectiveness of peer learning interactions, assuming that these systems are truly *supportive*—i.e., capable of coaching collaborating peers as they work on problems and critique other students' solutions. In order to achieve this goal, system developers need to know more about the types of coaching that students are typically able to provide to each other during problem-solving activities, and what types of coaching they need from more experienced students or teachers at various stages of their development in the instructional domain. System developers also need to know how *human* mentors provide such coaching, and—based on this information—to develop computer models of human guidance during collaborative interactions. The pilot study described in this paper aims to address these issues, and to assess the effectiveness of the research methodology we devised for doing so. We look forward to reporting on the findings that emerge from the data analysis phase of this work.

The students' dialogue contributions are flagged by an **Event** labelled STUDENT MESSAGE. The mentor's contributions are labelled as COACH'S MESSAGE. Dialogue text is in *italics* and is unedited. Comments are in **boldface**. Ellipses signify deleted segments of the transcript.

...

****

The students reach an impasse (i.e., they are unable to help each other), and pose a question to their mentor (Brian).

Time: (2 May 1995 7:39:30 pm)
Event: STUDENT MESSAGE
Text: *"bryan, do we assume that the relay card is getting pwr [sic; should read "power] or would it be a good idea to test it?"*

****

The mentor replies.

Time: (2 May 1995 7:39:51 pm)
Event: COACH'S MESSAGE
Text: *"NEVER assume anything!!! you should verify that the card is bad by checking inputs, outputs, and data controls signals"*

****

...

The students set up the handheld meter, and make a few measurements.

Time: (2 May 1995 7:42:39 pm)
Event: MEASUREMENT taken.
Device: HHM
Red Probe: 33
Black Probe: 30
Reading: 0.0000 Vdc

****

Figure 1: Computer-generated transcript of a collaborating dyad communicating with a mentor across networked machines

References

1. Baker, M., & Bielaczyc, K. (1995). Missed opportunities for learning in collaborative problem-solving interactions. Paper submitted to the *1995 World Conference on Artificial Intelligence in Education.*
2. Brown, A.L., & Palincsar, A.S. (1989). Guided, cooperative learning and individual knowledge acquisition. In L. Resnick (Ed.), *Knowing, Learning, and Instruction: Essays in Honor of Robert Glaser.* Hillsdale, NJ: Lawrence Erlbaum Associates.
3. King, A. (1989). Verbal interaction and problem-solving within computer-assisted cooperative learning groups. *Journal of Educational Computing Research, 5(1)*, 1-15.
4. Lesgold, A., Eggan, G., Katz, S., & Rao, G. (1992). Possibilities for assessment using computer-based apprenticeship environments. W. Regian & V. Shute (Eds.), *Cognitive approaches to automated instruction* (pp. 49-80). Hillsdale, NJ: Lawrence Erlbaum Associates.
5. McManus, M.M., & Aiken, R.M. (1993). The group leader paradigm in an intelligent collaborative learning system. In P. Brna, S. Ohlsson, and H. Pain (Eds.), *Proceedings of the World Conference on Artificial Intelligence in Education* (pp. 243-256). Edinburgh, Scotland, August, 1993.
6. Palincsar, A.S., & Brown, A.L. (1984). Reciprocal teaching of comprehension-fostering and comprehension-monitoring activities. *Cognition and Instruction*, 1(2), 117-175.
7. Slavin, R.E. (1990). Research on cooperative learning: Consensus and controversy. *Educational Leadership, 47 (December 1989/January 1990)*, 52-54.
8. Webb, N.M. (1987). Peer interaction and learning with computers in small groups. *Computers in Human Behavior, 3*, 193-209.
9. Webb, N.M. (1989). Peer interaction and learning in small groups. *International Journal of Educational Research, 13*, 21-39.
10. Webb, N.M. & Farivar, S. (1993). Promoting helping behavior in cooperative small groups in middle school mathematics. *Paper presented at the 1993 American Educational Research Association.*

Author's Address

Sandra Katz: 521 Learning Research and Development Center, University of Pittsburgh, Pittsburgh, PA 15260. katz+@pitt.edu.