Observations of Collaborative Behavior in COMPS Computer Mediated Problem Solving

Jung Hee Kim
North Carolina A&T State U.
jungkim@ncat.edu

Melissa Desjarlais
Valparaiso U.
melissa.desjarlais@valpo.edu

Kelvin Bryant
North Carolina A&T State U.
ksbryant@ncat.edu

Michael Glass
Valparaiso U.
michael.glass@valpo.edu

Abstract

COMPS is a web-delivered computer-mediated problem solving environment for student collaborative exploratory learning. The primary mode of interaction is typed dialogue, but COMPS also provides problem-specific affordances for exploring a problem. This paper reports qualitatively on dialogues from students employed in four different activities: two logical reasoning problems in a quantitative literacy class and two different problems in object-oriented Java in an elementary programming class. In all domains we observe behaviors consistent with quality collaborative learning experiences: co-construction of knowledge, mixed initiative dialogue, coming to common agreement, and students adopting different roles in the problem-solving process. These observations confirm that COMPS indeed facilitates true collaborative activity.

Introduction

COMPS is a web-delivered computer-mediated problem solving environment designed for supporting problem-solving activities in mathematics and computing [Desjarlais, Kim, Glass, 2012]. What the student mainly sees is a chat interface. For some problems, it has specific problem-related affordances for the students to manipulate. COMPS shows the instructor the conversations in real time, permitting the instructor to intervene. It records all events for analysis.

The goal of the COMPS project is to provide a computer-aided instrument for collaborative learning of concepts through problem-solving dialogue. We anticipate that the computer will aid the instructor, who is effectively looking over the shoulders of the students as they work, by providing a status display of progress toward solving the problem and degree of cooperative behavior.

This paper illustrates examples of dialogues collected using COMPS. Usage to date has been for testing, revising and refining the prompts for the problems and protocols, as well as collecting data in preparation for developing computer monitoring technology.

Thus there are two categories of phenomena that we look for in our dialogues. A) Observable instances of students making their thinking visible are necessary for computer monitoring. B) Observable examples of students helping each other learn are necessary for validating that the dialogues are facilitating collaboration.

The purpose of this paper is to illustrate examples of these phenomena as observed in our class usage.

Background

The student skills that are the focus of this project are oriented toward understanding and manipulating concepts. This is what [Skemp, 1987] calls "relational understanding," as a complement to the instrumental skills of programming that are the bread and butter of the elementary programming classes or the algebraic skills that are the bread-and-butter of elementary mathematics classes.

Collaborative problem solving directly addresses the goal of helping students learn from each other. Our approach so far has employed the theory of group cognition studied by the Virtual Math Teams project, where students solve math problems through computer-mediated chat [Stahl, 2009]. In group cognition different members of the conversation provide statements that, if they were uttered by one person, would be taken as evidence of a cognitive process. Stahl presents an attested example of group cognition in a team working on an algebra problem [2009, pp. 57-73]. In Stahl's example, different members of the team mooted ideas into the conversation but could not solve the problem. Stahl shows that you can thread through these ideas, building a correct analysis of the problem, until one student provided an answer that was predicated on the previous thoughts but did not explicitly refer to them. Possibly because the final answer did not explicitly refer to the other students' previous utterances, the other members of the team all attributed the solution to the student who provided the ultimate answer. But Stahl shows that a team cognition analysis makes more sense.

We observed similar examples of team cognition in our observation of students studying a nim-like game in small groups in person [Dion, Jank, and Rutt, 2011]. The bits of realization comprising a solution path were mooted by different people. Sometimes one realization is explained or defended by a person different than the one who expressed it.

There is also research showing that collaborative activity is a desirable pedagogical approach specifically for creating conceptual understanding [Tchoukine et al., 2010]. Key to engendering learning is dialogue that engages in domain reasoning, such as explaining, negotiating, or inferring [Stahl, 2004]. Justifying, arguing, and similar knowledge-engendering dialogue moves were notable in the VMT dialogues [Zhou, 2009].

Regarding our goal of making thinking visible, discourse pragmatics provides the theoretical justification that it should occur in problem-solving dialogues. Koschmann's studies of doctors in training [Koschmann, 2011] show that not only do participants articulate explicitly, they are also obligated to communicate their level of understanding as part of grounding. Discourse obligations are mostly socially-derived behavioral expectations, such as taking turns and answering questions. Grounding is the obligation to achieve common understandings [Clark and Brennan, 1991].

Following Grice's Cooperative Principle [Grice, 1975] the important discourse obligation we observe in these dialogues is to make sure everybody is aware of your knowledge state. We claim that if you are actively participating in a problem-solving dialogue, and you don't signal your knowledge state, Grice's maximum of quality permits the implicature that you have a state approximately the same as everybody else. It would be pragmatically odd to continue listening or participating in the dialogue without communicating to the others that you have figured it out when the others haven't. Similarly it would be odd to let the others continue on to the next problem without signaling that you still do not understand the current one.

This often holds in ordinary situations, however in pedagogical situations there are alternative explanations for a lack of grounding. For example, students may save face by not revealing their lack of understanding. One question our data will answer is how often we observe students working to achieve common understanding.

COMPS Usage

We have used COMPS with four problems in classes at NC A&T State and Valparaiso Universities. These problems were administered as approximately one hour regular class exercises in a computer lab, with the students deliberately seated far from each other. Collaboration groups were typically four students, but could be as large as six. The four problems are:

1. The Poison problem, a logic exercise usually assigned in a quantitative literacy class. Students must figure out the winning strategy for a Nim-like game where players remove one or two stones from a pile. The person to remove the last stone loses. In these exercises, the students have been elementary education majors attending the mathematics content course. We have 19 sessions from three class sections.
2. The Patagonian Congress problem, an exercise in voting mathematics. Students must figure out how proposals pass or fail in the congress under various voting procedures and numbers of people ranking the proposals in different orders. This is another quantitative literacy problem assigned in an education methods class. We have 5 sessions.
3. A puzzle in Java programming language class inheritance (Figure 1). Students must figure out the inheritance relationships between four different Java classes, based on the output of a method call. Students were in the second semester programming class. We have 13 sessions from two class sections.
4. A puzzle in Java Swing programming, deriving program and class structural relationships from the screenshot and functionality of a simple graphical user interface. Students were also in the second semester programming class. We have 10 sessions from two class sections.

The quantitative literacy exercises have on-screen manipulatives. Playing the Poison game is configured to encourage experimentation rather than competition. For example students can play any side without dividing up into teams. The Java problems use COMPS as a chat interface only.

In the quantitative literacy protocols the students are required to meet in-person as a group after about three-quarters hour of online work in order to write up their results. This mimics the normal face-to-face classroom administration of these problems. But in our experiments it meant we did not capture the end of the collaborations and their summarized results, which happened in-person. In the programming protocols one student from each group obtains the correct answers from the professor after the group agrees on an answer. Then, continuing to work online, the group has to explain the correct answer. This protocol has the effect of forcing the one student to communicate online to the others what he or she has learned, but we don't see what the professor said.

Illustrations of Collaborative Behaviors

Group Cognition

Figure 2 illustrates a dialogue in class inheritance. This example is noteworthy because it is visibly a group cognitive process where all the students participated, but they participated through adopting different non-overlapping roles. For this analysis we are have annotated the dialogue turns as Initiate or Response dialogue moves, in the analytical model of Conversation Analysis. [Wells, 1999].

- Student A takes the metacognitive role.
- Student B, who does most of the work in finding the answer, is almost entirely directed by A.
- Student C, who participates the least, is the one who says the answer first.

First let's look at student A. For the first 13 turns the only initiator is A. Every other utterance is in response. We notice that A performs the following functions. Items 1 through 4 are arguably regulating the group cognitive process, which is the basis of our claim that A is taking a metacognitve role.

1. Gaining agreement on everybody's knowledge state (turn 1)
2. Setting the problem-solving agenda (turn 4)
3. Deciding when to summarize or articulate the state of knowledge so far (turns 6, 12, 18)
4. Plausibility checking the current state of knowledge, noticing it has gone off track (turn 8)
5. Suggesting an analogy that might be part of the solution (turn 15).

Turn 15 was A's only direct contribution to the evolving knowledge state, and it was not relevant.

Now let's look at student B, who provided the reasoning. In turns 5, 7, 11, and 13 this student develops and articulates the solution. When student A demands an explanation at the end, student B provides it. B's other turns are all dialogue responses. Student B is the one who is most willing to articulate the solution.

Finally we look at student C, who contributed only three turns, the fewest. At the beginning (turn 3) C does not know the answer. As of turn 14 C still does not have the answer but contributes knowledge. In turn 16, the student says the correct letter from the multiple-choice answer. In other words, even though C lurked without contributing to the conversation for 10 turns, there is strong evidence that C was engaged.

Obligation to include everybody

Figure 3 illustrates a dialogue solving the Poison problem. This example is noteworthy because it illustrates the power of discourse obligation, and how those obligations serve our twin goals of collaborative learning and making thinking observable. In this dialogue there is no one student setting the agenda as in the first example. And no student demands an explanation so as to achieve the same level of enlightenment as the others. However the evidence is that the students were attending to each others knowledge state anyway.

Much of the activity in the Poison dialogues is concerned with repeatedly playing the game multiple times, with occasional analysis. In this extract we pick up at dialog turn number 124, the first moment when student C figured out the first insight. She articulated this insight for everybody.

But not everybody understood the point. They played another game so C could show the others. Then they played more games, progressively smaller to make the point more evident. Student B interrupted a game to point out and explain the winning strategy to the remaining two students (turns 165 to 177).

The process of ensuring that the whole group achieved a common level of understanding, starting from turn 124, took 16 minutes out of a one hour lab. Students B and C knew that this first realization was not the entire solution, they could have proceeded to finish the problem. Ultimately they sacrificed their opportunity to finish the problem in order to aid the other two students.

Discussion and Conclusions

Evidence that our students are actively participating in group cognition is easy to find in our dialogues.

The Java inheritance exercise in Figure 2 shows that the [Kersey, et al., 2009] experiment in initiative is incomplete. That study of programming problem collaborative dialogues found that taking the initiative correlated highly with learning gains. This held true for both dialogue initiatives (controlling discourse focus, e.g.) and task initiatives (coordinating problem-solving tasks, e.g.). In our example above, student A both controlled both the conversational focus and the problem-solving agenda, yet this student had arguably the weakest grasp of the answer.

Evidence of both student knowledge and knowledge state is visible in almost all of our transcripts. The students indeed make knowledge visible as [Koschmann, 2011] predicts.

The primary factors that explained when our dialogues lacked evidence of collaboration were:

- Some of the Java problems were too easy to solve. In many groups there was little evidence of knowledge co-construction. Somebody simply said the answer.
- Playing the Poison game can be too much fun. Groups can expend a lot of time simply playing the game without addressing the problem statement [Desjarlais, Kim, Glass, 2012].

- When playing the Poison game, some groups became competitive. Individual participants have been observed refusing to divulge their insights in order to keep the advantage when playing trial runs of the game.

These factors can be addressed by changes to the protocols and the COMPS environment.

These preliminary sessions show that COMPS usage is consistent with our goals of a) facilitating high-quality collaborative problem-solving, and b) producing visible evidence of student knowledge state and cooperative behaviors that the computer can potentially monitor.

Acknowledgements

Thank you to our hard-working students at North Carolina A&T and Valparaiso Universities. This work is supported by the National Science Foundation under awards 0851721 and 0634049 to Valparaiso University and 0633953 to North Carolina A&T State University.

References

Clark, Herbert H., and Susan E. Brennan. 1991. Grounding in Communication. In Resnick, L. B.; Levine, J. M.; Teasley, J. S. D., *Perspectives on Socially Shared Cognition*, American Psychological Association.

Desjarlais, Melissa, Jung Hee Kim, and Michael Glass. 2012. COMPS Computer Mediated Problem Solving: A First Look. In Proceedings of the Midwest Artificial Intelligence and Cognitive Science Society Conference (MAICS 12), Cincinnati.

Dion, Lisa, Jeremy Jank, and Nicole Rutt. 2011. Computer Monitored Problem Solving Dialogues. Report of 2011 VERUM summer REU. Department of Mathematics and Computer Science, Valparaiso University, Valparaiso, IN. Retrieved March, 2013. http://www.valpo.edu/mcs/pdf/reu2011glasspaper.pdf

Kersey Cynthia, Barbara Di Eugenio, Pamela Jordan, and Sandra Katz. 2009. Knowledge Co-construction and Initiative in Peer Learning Interactions. In *Proceeding of the 2009 conference on Artificial Intelligence in Education: Building Learning Systems that Care*. IOS Press.

Koschmann, Tim. 2011. Understanding Understanding in Action. *Journal of Pragmatics*, 43 (2) pp. 435-437.

Skemp, Richard. 1987. *The Psychology of Learning Mathematics*, Hillsdale, NJ: Erlbaum. Chapter 12.

Stahl, Gerry. 2004. Building Collaborative Knowing: Elements of a Social Theory of CSCL. In J. W. Strijbos, P. Kirschner and R. Martens, eds., *What we know about CSCL: And implementing it in higher education*. Boston, MA: Kluwer Academic Publishers., pp. 53-86.

Stahl, Gerry, ed. 2009. *Studying Virtual Math Teams*, 2009, Springer.

Tchounikine Pierre, Nikol Rummel, and Bruce M. McLaren. 2010. Computer Supported Collaborative Learning and Intelligent Tutoring Systems. In R. Nkambo, J.Bourdeau, & R. Mizoguchi, eds. *Advances in Intelligent Tutoring Systems*. Springer. Chapter 22, pp. 447-463.

Wells, Gordon. 1999. *Dialogic inquiry: Towards a Socio-cultural Practice and Theory of Education*. Cambridge University Press.

Zhou, Nan. 2009. Question Co-Construction in VMT Chats. In Stahl, 2009, pp. 141-159.

What should blanks 1 – 4 contain to produce the following output:

From Foo_3 From Foo_2 From Foo_1 From Foo_3 From Foo_4 From Foo

```
public class Foo {

    public static class Foo_2 extends _______ { // 1
        public Foo_2() {
            System.out.print("from Foo_2 ");
        }
    }

    public static class Foo_1 extends _______ { // 2
        public Foo_1() {
            System.out.print("From Foo_1 ");
        }
    }

    public static class Foo_4 extends _______ { // 3
        public Foo_4() {
            System.out.print("From Foo_4 ");
        }
    }

    public static class Foo_3 extends _______ { // 4
        public Foo_3() {
            System.out.print("From Foo_3 ");
        }
    }

    public static void main(String[] args) {
        Object foo_2 = new Foo_1();
        Object foo_3 = new Foo_4();
        System.out.println("From Foo ");
    }
}
```

Possible Answers:

a) 1) Object 2) Foo_1 3) Foo_2 4) Foo_3
b) 1) Foo_4 2) Foo_2 3) Object 4) Foo_1
c) 1) Foo_3 2) Foo_2 3) Foo_3 4) Object
d) 1) Foo_3 2) Foo_2 3) Object 4) Foo_3
e) None of the Above

Figure 1. Problem in Java inheritance.

Turn	Stu		I/R	
1	A	do you guys understand this second problem	I	Most initiate-response pairs in this group are initiated by A.
2	B	this one is confusing.	R	
3	C	yeah this one got me thinking	R	
4	A	lets try and take it like one output at a time...how are we gong to get this to print Foo_3 first? [*ellipsis dots in the original*]	I	A sets the problem-solving agenda
5	B	we need to first make foo_2 extend foo_2	R	B moots first important idea
6	A	why	I	A prompts for explanation
7	B	because foo_2 starts the main method but it isnt the first thing that prints	R	B explains
8	A	wait hold on..that cant be right its not a choice bro so it has to start with foo 3 or 4 or object	I	A notices the multiple-choice answers do not include the proposed answer.
11	B	oh that's what i meant . we have to make foo_2 extend to foo_3 my bad	R	B corrects first idea
12	A	so when you do foo 2 extends foo 3 , the program goes down to foo 3 and prints out "From Foo_3"?	I	A articulates B's idea more fully
13	B	yes and then it goes back to foo_2 to print "From foo_2" .	R	B finishes
14	C	so what is the main calling when it says Object foo_2 = new Foo_1? and for the other	I	C shifts focus to next part
15	A	idk it kinda looks like a swap without the "temp" thing/example Dr. <Instructor> showed us	R	A provides not-relevant analogy
16	C	I got answer c	I	C is first to provide correct answer
17	B	i do too.	R	B concurs
18	A	can you explain it to me because i am confused	I	A prompts for explanation
19	B	ok , i got it now. (*types first part of explanation, 42 words. Subsequent dialogue elicits rest.)*	R	B explains all

Figure 2. Distinct roles in class inheritance problem.

Turn	Stu	Utterance
		(A takes 1)
122	D	good choice
123	A	we're gonna lose :(
		(D takes 2, wins)
124	C	i figured it out yooo. [...deletia...]
126	B	ok let's talk about it now lol
130-131	C	Ok, So does anyone else know the trick? Or am I just awesome?
132	B	idk
133	Prof	What's the trick?
134	C	Is it to get the opposing team on the fourth tile? it's something to do with the number 4 because that's how we won each time haha
135	D	but that only works if we get four tiles when it's our turn. how do we guarantee that we get that?
136	Prof	The number 4 is special, but do you want 4 on YOUR turn or on your OPPONENT'S turn?
137	A	can we play again?
138	B	yeah with 10
		(play several games, with explanations between some of them)
158	A	i still don't get it at all..
159	B	we didn't have four this time
160	D	haha true
161	C	i'm going to restart it at 5
		(more playing)
163	D	so what are we deciding? do we want the opponent to have four or our own team?
164	A	idk lets do it with five again..
165	B	we need to set it up so we have 4
166	B	so whoever is first take 1
167-168	D	so whoever goes first only choose one. great minds think alike!
169	B	:)
170	A	cbad (this is their convention for turn-taking order of game playing.)
		(D takes 1)
171-176	B	ok so ithere is four for me and A. you guys set up 4 for the opponent....
	B,A	(Illustrate and explain)
177	B	so the trick is. you want to set up 4 for your opponent
178	C	CORRECT!
179	B	yayyy! we figured it out

Figure 3. Obtaining common understanding in Poison.

'