Understanding Pair-Programming from a Socio-cultural Perspective

Weiqin Chen, Marius Nordbø
Department of Information Science and Media Studies, University of Bergen
POBox 7800, N-5020 Bergen, Norway
Email: weiqin.chen@infomedia.uib.no, marius.norbo@student.uib.no

Abstract: Pair-programming has been used more and more often in educational settings. Although some studies have been conducted and pair-programming was found in these studies more effective for students to learn programming, very few studies have focused on the interactions between the paired students during the pair-programming process. The study reported in this paper seeks to understand the learner's activities by taking a closer look at the interactions between the pair from a socio-cultural perspective.

Introduction

Pair-programming is one of the 12 practices in XP (Beck, 2000) where two persons sit in pairs working on a problem apposed to a more isolated single-programmer situation. One person, the driver, is in control of the keyboard or pencil, while the other, the observer, actively observes the driver. The observer uses resources to find better solutions, watch for defects in the code and together they collaborate and discuss problems they encounter and in that way knowledge constantly is shared between them. They can periodically switch positions and both work towards a shared goal. Beck (2000) emphasizes that pair-programming is not an activity where one is doing the work and the other is just watching. Pair-programming is a dialog between two persons trying to program together at the same time and together understand how to program better.

Research shows that using pair-programming in an educational setting is effective for teaching students how to program (Preston, 2006; McDowell et al., 2003; Williams & Upchurch, 2001) and it results in higher quality programs, greater understanding of programming process, decreased dependencies on teachers, improved course completion rates and scores, and decreased time to complete programs. However, most of the research take a quantitative method and focus on a rather big number of students in a programming course. Very few researches have studied intensively the interactions between members of the pair during the pair-programming process. The study reported in this paper seeks to understand the learner's activities by taking a close look at the interactions between the pair from a socio-cultural perspective.

Experiment

The experiment was conducted over a period of four days. Eight students in a first year introductory course in object-oriented programming participated in the experiment – seven males and one female. The students were beginners regarding programming and object-orientation although their skills and experiences differed. They were paired up forming four groups and scheduled to meet at the research location one group each day. Each session lasted approximately four hours including one break.

The experiment was conducted in a room with a table and two chairs. One laptop was set up with an external keyboard and a mouse. The main focus in this experiment is to capture the collaboration between the participants and how they interact with each other. The camera was therefore positioned in front of them. From this angle the screen was not captured. This was resolved by streaming pictures of the screen onto the laptop. Both video and screen pictures were streamed on to a hard drive. A large extent of the activity in the room was framed by these two views. The teachers/researchers were situated in the next room observing all the activity on monitors and they could at any time enter the room of the experiment. They could initiate contact when the participants had problems with the software or when they were stuck on a task. Participants could also initiate the contact by asking for help. The camera recorded all activities between teachers/researchers and participants. The participants were presented a written assignment text including a set of four tasks. They also received 600 lines of Java code with initial functions as a starting point, both printed version and electronic version, and a graphical sketch of the data structures in the program they were to work on.

Data Analysis and Findings

In this study the research focus is on collaboration between students in a learning situation. A number of guiding questions were identified and interesting segments from the video were analysis accordingly. In this paper we focus on one guiding question: *How do the pair-programming rules (especially the driver/observer rules) affect the interaction between paired students?* According to Vygotsky (1978) a learner should have constraints (e.g. rules) to help them interact with his or her surrounding elements, but constraints and freedom must be balanced to create a constructive situation for the person. In pair-programming member's activities are strongly influenced by the role she/he takes.

In the following segment the participants in Group 3 are working on Task 2 and the method finnLedigFirma. They have discussed the assignment and possible solutions, but they are uncertain and decide to consult the assignment text before they continue working on the problem. In the following segment the observer suggests using the variable 'kapasitet' in order to find the firm with free capacity (line 186 and 188), but the driver doesn't follow up on the proposed idea. The driver continues with his own ideas and the observer follows his suggestions and agrees with them. They continue talking in turns and the driver types what they agree on.

Line	Time	Activity	Actor	Conversation
184	00:30:00	Reaches for the assignment text	Driver	Let's see the assignment text
185	00:30:13	Reads from the text	Driver	Find and select the first Byggefirma with free capacity
186	00:30:25	Looks at the printed code	Observer	Then we have to check the 'kapasitet' thing also
187	00:30:27	Confirming	Driver	Yes
188	00:30:30		Observer	We can check the capacity… And then…
189	00:30:38	Interrupts the observer	Driver	We must have the variable 'funnet' and then we can have a while-loop which checks on the Byggefirma… the firmaliste…
190	00:30:55	Looking at a code segment in the printed code	Observer	If we use this one… Hmmm… Then we can just set a boolean…
191	00:31:05		Driver	Yes
192	00:31:10		Observer	We must have one that limits the method so we don't put in where there is no capacity. Right?

From line 221 the observer again tries to suggest using the variable 'kapasitet' (see line 186). But the driver still ignores it and now he is no longer responding to the observer. The communication breaks down and the driver types without consulting his peer. The observer then becomes frustrated and he takes control over the keyboard and types in his suggestion (line 225). This action forces the driver to think carefully about what the observer has suggested. From the later actions, we can see that the driver ends up with actually following the observer's suggestion. They continue collaborating and end up with a solution that is very similar to what was written in line 225.

Line	Time	Activity	Actor	Conversation
221	00:39:30		Observer	But it must be with byggefirma dot… Or… It's a different class… Or has it an array?
222	00:39:50	Without answering he types in more in the method	Driver	Funnet = true; And we need an else if.
223	00:40:16		Observer	Yes, that's if it is zero? Or?
224	00:40:17	Doesn't answer	Driver	
225	00:40:22	Takes the keyboard and types his suggestion	Observer	I have to write this…
226	00:40:50	Looking at the screen before he takes control over the keyboard again and deletes what the observer wrote	Driver	Oh! You then must search in the firm's antall
227	00:41:20	Studying Byggefirma class	Driver	Firmaliste.søknader, then
228	00:41:36		Observer	Yes, antall. What I mean is that if antall is

				equal or larger than 1 and less than capacity… You see?
229	00:42:27		Driver	Wait a minute…
230	00:42:58		Observer	Yes, we have to look at when they have one or more applications in…. Can we use many AND in a statement?
231	00:43:10	Types in code similar to what the observer did in line 225	Driver	
232	00:43:10	Saves file and compiles. It seems to work.	Driver	Yes, I think so. Let's see if what we have is ok.

The driver/observer rule, where the driver has control over the keyboard while the observer is commenting and participating in the discussion of what should be written, is made explicit through the psychical restrictions that the two students only had one keyboard and it enforces an equal division of labor in the group. According to Kuutti (1996), when members of a community follow the same rules they are able to work towards a common objective and the embedded rules support the communication on an operational. When participants fail to follow the same rules a contradiction appears and harms the communication between them. In order to restore communications the conflict must be resolved by visualizing the rules and make all members aware of them. The rules are then supporting the communication on an action level. In the above segment a contradiction appears. The observer suggests a solution (line 186) but the driver ignores him. After a while the observer expresses the same solution only to get no reaction from the driver. The rules are broken since the driver does not listen to the observer (line 221 – 224) and the activities towards their objective (finishing the assignment) is harmed by it. To restore the communication the observer "oversteps" the guiding driver/observer rule and takes the keyboard (line 225). With this action he makes the driver shift his focus from his own individual work to the social collaboration between the two students and he makes visible that they are both equally connected to the task they are working on. When the rules are visible they support the communication on an action level and this restores the relation between them. As an outcome of this contradiction the students came closer to completing the task since the observer's opinion was.

The activities in the above segment show that it is necessary for the students to follow the pair-programming rules and that it helps the communication between them. When they are aware of each other and have equal influence in the decision-making process, they are able to work together towards finishing the assignment. If one of them does not include the other in their problem-solving he or she needs to be reminded. In the segment above we see such a contradiction occurs and the participants are able to solve it by themselves.

Conclusion

The goal of this research is to understand learners' activities in pair-programming process. We analyze the activities from four pair-programming processes. The focus of this paper is the effect of driver/observer rule on the interactions. The analysis indicates that in pair-programming, the driver/observer rule facilitates collaboration by providing pairs with opportunities to think aloud and reflect. However, this does not mean that pair-programming is automatically a success. Constructive pair-programming depends on the peers' engagement (to the process) and how well they comply with the guiding principles.

References

Beck, K. (2000). *Extreme Programming Explained: Embrace Change*. Addison-Wesley, Boston.

Kuutti, K. (1996). Activity Theory as a potensial framework for human-computer interaction research. In B. Nardi (Eds.). *Context and Consciousness: Activity Theory and Human Computer Interaction* (pp. 17-44). Cambridge: MIT Press.

McDowell, C., Werner, L., Bullock, H., & Fernald, J. (2003). The Impact of Pair Programming on Student Performance, Perception, and Persistance, *Proceedings of the 25th International Conference on Software Engineering (ICSE 2003)*, pp. 602 - 607, May 3 - 10.

Preston, D. (2006). Using collaborative learning research to enhance pair programming pedagogy. *ACM SIGITE Newsletter, 3(1).*

Vygotsky, L. (1978). *Mind in Society: The Development of Higher Mental Processes*. Cambridge, MA: Harvard University.

Williams, L. & Upchurch, R. (2001). In Support of Student Pair-Programming. *Paper presented at SIGCSE Conference on Computer Science Education*, Charlotte, NC, February.