Manuscript received Jul. 1, 2007; revised Sep. 11, 2007

Online Pair-Programming for Learning Programming of Novices

SOOJIN JUN†, SEUNGBUM KIM†, WONGYU LEE††
Computer Science Education, Graduate School†,
Dept. of Computer Science Education, College of Education††
Korea University
Anam-Dong, Sungbuk-Gu, Seoul,
KOREA
soojin.jun@inc.korea.ac.kr†, seungbum.kim@ inc.korea.ac.kr†, lee@comedu.korea.ac.kr††

Abstract: - The purpose of this study is to determine the effects of online pair programming for novices. This paper describes two system designs for online pair programming: real-time synchronous pair programming and non-real-time asynchronous pair programming. An experiment was conducted to test the feasibility of online pair programming, using Squeak eToys as an educational programming language. Nebraska of Squeak eToys, a toolkit for building remote interactions with morphic (object-oriented and visual) projects in Squeak eToys, was used for sharing learners' screens and input devices, such as the keyboard and mouse. In this experiment, the following were compared: co-located pair programming, distributed pair programming, and individual programming, through the statistical analysis of their task scores; and a number of creative ideas and their scores in the final test, using T-test, ANOVA F-test, and post-comparison. The results showed that distributed pair programming is not significantly different from co-located pair programming, and that pair programming is more effective than individual programming. In addition, based on the students' sentiments and class observations, several suggestions are given herein, such as environments or teaching methods in online pair programming. The importance of communication, authority control, and peer pressure in online pair programming is also discussed in this paper. In conclusion, the results of this experimental research show that educators can be helped in designing effective learning systems for online pair programming.

Key-Words: - Programming Learning, Pair Programming, Squeak, Novices, On-line Learning

1 Introduction

The computer and information technology (IT) revolution has given birth to numerous innovative ideas that have dramatically changed human lives. New ideas are born, and existing ideas become a reality, with the aid of computers. In this era, where IT is an important resource, the interest and investments in computer education in the K-12 curriculum are constantly increasing. As such, appropriate computer education methods and tools are desired, especially in the area of computer programming, which is an essential part of the computer science.

Syntaxes and logics can serve as significant stumbling blocks for novices in computer programming learning [1]; these can prevent them from fully learning computer programming [2]. Many researches have thus been conducted to improve the learning environment of computer programming. An efficient collaborative learning method that was developed from such research was pair programming (PP), which was originally intended for use by commercial programmers. PP is widely acknowledged to be more effective than individual programming, which was developed by extreme programming (XP) developers and researchers [3]. It can finish a given task twice as fast as individual programming can, and it is known to produce higher-quality computer codes. Moreover, PP is known to improve team communication skills and to portray computer programming as enjoyable rather than burdensome [4, 5, 6]. Pair programming, therefore, is a useful and effective programming

education method. More PP learning environments, however, must be developed, as collaborative learning environments can overcome the temporal and spatial limits of learning and can thus lead to greater learning among learners in different places, of different ages, and of different personalities.

The effectiveness and features of PP are presented in this paper, along with the results of the experiments that were conducted to determine the feasibility of distributed PP for distance learning, to compare co-located PP with distributed PP, and to determine the effects of PP by comparing the scores of the learners in PP classes with those of the learners who performed the same tasks individually.

2 Background

2.1 Pair-Programming

The Pair-programming is a style of programming that two programmers work side by side at one computer, continuously collaborating on the same design, algorithm, code, or test [7]. The PP could be used in classroom with several synergistic behaviors. Students put a positive form of pressure on each other. It allows pair negotiation, reviewing and debugging together to get the best solution. Knowledge is continuously being passed between partners. The students feel much more fun with PP more than working alone [6].

2.2 Related Works

There have been many studies on the PP. It is found that the PP improves design quality for the development-time cost of about 15%, and, reduces defects, and staffing risk, while enhancing technical skills, and improving team communication. Additionally it is considered to be more enjoyable for students at statistically significant levels than programming independently [4, 5, 8].

Moreover, Baheti indicated that the distributed PP is a feasible and efficient method for dealing with team projects [7]. H. Shen showed that the internet-based real-time collaborative programming allows physically dispersed programming teams to concurrently and collaboratively design, code, test, debug and document the same program [9].

3 Online Pair-programming Systems

In this section, we propose the system for an online pair-programming. Even if an asynchronous pair programming is slightly inefficient, we present not only a synchronous pair-programming system but also an asynchronous pair-programming system.

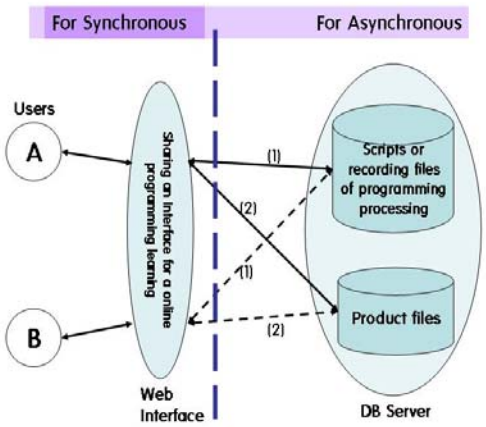

Fig. 1 Pair-programming for Synchronous and Asynchronous in online programming learning environment

Firstly, in case of the synchronous situation that distributed learners A and B program on real-time, the learner A and B can perform an assignment at the same time by a rule. At that time, the system must allow to program at the same time with sharing their working interface and support various communication devices for communicating smoothly like internet chatting applications, white board and web board. In addition, while one user works, it is necessary to control so that others and a user cannot work.

Second is the asynchronous situation that distributed learners A and B program on non real-time. A program is constructed by the system, which exchanges the authority for carrying out a subject after students have deliberations to a subject with putting in order. The learner programs for a set time then the system store a file recoded or scripts included all of programming process. When learner A finishes to program, the system stores a final programming product as well.

In the same way, the learner B can connect after the learner performs. When the learner B connects, the system applies the stored files recoded or scripts included all of programming process of the A. Next the learner B looks at the programming processing, he programs taking over to the product file loading from the DB server.

As they repeat the process like that, the learners A and B could get an effect of pair-programming in the asynchronous learning environment as well.

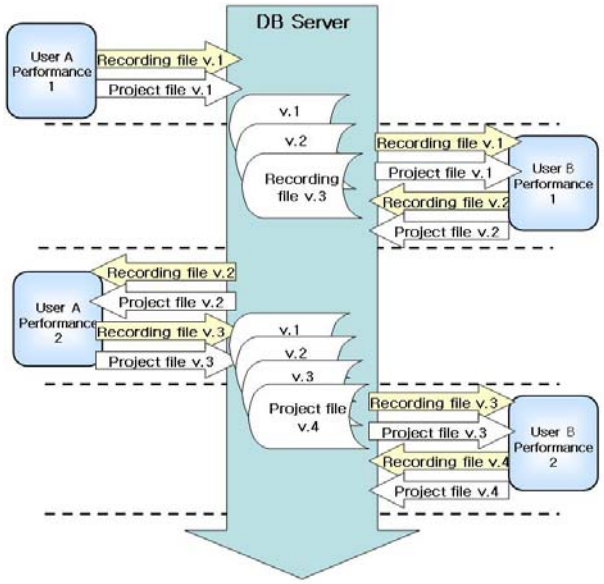

Fig. 2 Learning Processing for Asynchronous Pair-programming

4 Experiment

As we have seen, existing researches have presented an effect on the PP in off-line or on-line environment for commercial programming. Therefore, this paper makes an outline of experiments about the feasibility of online PP for novices who learn programming as the subjects for general education. We presumed that the learning effect of the online PP would be similar to that of the offline one.

Especially, we had an experiment with Squeak using Nebraska for the distributed PP. Nebraska is a toolkit for building remote interactions with morphic (object oriented and visual) projects in Squeak e-toy. The students can interact with each other using input device like a mouse or a keyboard in same contents of screen [10]. Nebraska allows performing real-time programming tasks by sharing a interface simultaneously.

In the experiment, we compared the learning effects among a co-located class, a distributed class, a individual class. It was performed during five weeks (10 hours, 2 hour a week), for 3rd grade students (Three class, N=106) of National University of Education in Korea during May ~ June 2006. The students have never learned about the computer programming. Then, they performed four various tasks with 3 methods for learning programming, and had individually an final test at the 5th week We supposed that their learning ability and interest were equivalent for the students of the same university and the same grade. A curriculum for the experiment was shown in Table 1.

Table 1 Curriculum for learning programming

H	Theme	Main Concepts	Notes
1	What is Squeak? Drive a car.	Interface, Drawing, Objects, Making a Script, Loop statement	.
2	Drive a car using a joystick	Making a Script, Loop statement, Using a joystick	Task 1
3	The Lunar Lander Project	Variable, Conditional statement, Controlling the script	.
4	Apply an acceleration of the car	Variable, Controlling the script	Task 2
5,6	Car race	Conditional statement	Task 3
7,8	Making a simulation or a game	Control script, Conditional statement	Task 4
9. 10	Making a race simulation	Making a Script, Variable, Conditional statement, Controlling the script	Final test

The procedure of the classes is like figure 3. First, we let them follow the programming and understand the main concepts, and then I presented a task. Then, students solved tasks on pair-programming or individual programming. After about 10 minutes, we provided learners with hints on the tasks. And next, they solved the tasks again and submitted results online.

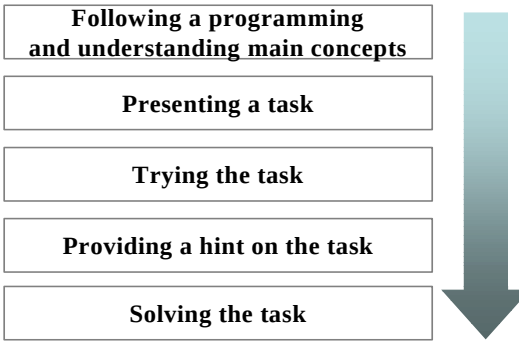

Fig. 3 Process of the learning programming

Three groups for this experiment are as follows:

The Co-located PP class (16 groups, n=32)

The first set was a co-located PP class. After the students were grouped in pairs, they performed the programming tasks side by side using one computer in the same space. There were no special conditions on

the system. They programmed by turns with an established rule and time on normal desktop computers.

The Distributed PP class (16 groups, n=32)

The second class was a distributed PP class. After the students were grouped in pairs, they performed programming tasks with interface sharing using each computer through a remote access by Nebraska [9]. At this time, we let them to chat with the group member through the network. They could use a messenger with a headset for smooth communication between each other.

The individual programming class (n=39)

The third class was an individual programming class. The student performed programming the tasks individually.

5 Result

As mentioned above, learners learned just the basic skills of programming and they were given tasks that a team can solve together. Then, at the final week, all of them had an examination individually. Especially, we compared the distributed PP class and the co-located PP class for the scores of tasks and the number of creative ideas as a quality on the tasks. Then, we compared to the final test among the three groups.

5.1 Scores of the Tasks

Learning the programming language effectively means that the learner can solve all the conditions of the task completely and elaborately. Accordingly in this experimentation, we scored all the items of the tasks in detail such as a naming objects and scripts; a using variable, conditional statements and a defining control buttons. The scores were compared using T-test as follows.

The program score of the initial tasks can be seen in Figure 4 as the distributed PP class had lower scores than the co-located PP one. This small difference, however, was not statistically significant with $p<.05$. This finding shows us that learners can learn by PP not only in an online learning environment but also in an offline one.

5.2 The creative idea

Another criterion for an effective programming learning is the quality of the completed program. Because of performing the tasks together, the learners can add new idea and the other scripts or functions. The creative idea is finding new algorithms or ways from adding various ideas beside the idea (of an algorithm) that the teacher presents at first.

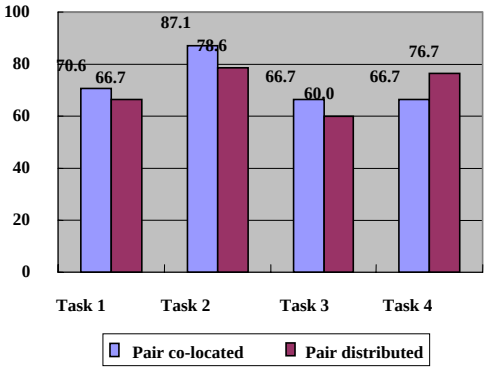

Fig. 4 Mean of Tasks' scores (%)

For example, when students make the project 'driving a car' on the motor-racing track, they used different number of sensors and colors (or the background), and other ideas like making some defenses, having acceleration about specific conditions.

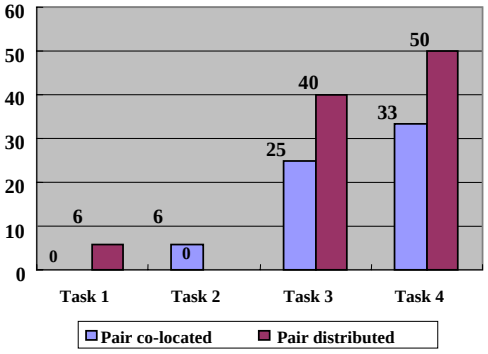

Fig. 5 The creative idea (%)

To sum up, it is evident that learners encourage each other and communicate smoothly while performing the given tasks. Figure 5 shows that both classes had various ideas as time passed. Especially, the distributed PP class had more creative ideas than the co-located PP class. This result shows that performing pair programming in distributed environments is not difficult due to share their ideas enough.

5.3 The Final Test

After performing all the tasks we gave them a final test. That programming task included all functions they had learned. The results came out as follows.

The individual programming class had the lowest score as Table 2. Furthermore, the programming

ability showed statistically significant difference by the programming methods (ANOVA: F=3.706, p<.05).

Table 2 Descriptive statistics of the final test scores

Methods	N	Mean	SD	SE
Pair co-located	32	6.6	1.07	0.19
Pair distributed	32	6.2	1.18	0.21
Individual	39	5.9	1.12	0.18
Total	103	6.2	1.15	0.11

Table 3 shows results of post-hoc comparison on the final test scores based on programming methods by using scheffe. According to the result of paired F test, only between pair co-located and individual method showed significant statistic (p<.05).

Table 3 Post-hoc multiple comparison by Scheffe (*p<.05)

(I) Methods	(J) Methods	(I-J)	SE	Sig.
Pair co-located	Pair distributed	0.438	0.281	0.30
Pair co-located	Individual	0.728*	0.268	0.03
Pair distributed	Individual	0.290	0.268	0.56

5.4 The Student Sentiments for the Classes

We made each student note a sentiment about the classes after they finished up program schedule. Some students in the distributed teams noted about the online PP education class as follows:

"It was fun that two people control together with a remote access.
"Because of performing together, we were full of good ideas."
"Because of distance-programming, though we didn't locate together. It was nice that we could cooperate."
"Although we were chatting, the communication was not easy. But sharing the interface and the input devices (mouse, keyboard), I think, it would be good if we would get used to the collaborative drawing."

We can see that distributed class's students also felt fun for the method. However, it is necessary to make some suggestions for online PP systems so that the students can use it much more easily and comfortably.

6 The Suggestions for the System

The following suggestions are hereby made based on the results of the test and of the class observations.

6.1 The Importance of Communication

The co-located-PP class was allowed to use not only fundamental communication methods but also various other communication methods, such as body language and the use of plan drawings or flowcharts. The distributed-PP class, on the other hand, was restricted to the use of particular communication methods; as such, there was a limit to the ideas that they shared and to their learning of the programming language. They could have shared more ideas and could have learned programming better if the system supported more convenient communication methods. Therefore, educators must support various communication methods, such as the use of a white board and visual chatting, so that the learners in a class can share ideas effectively among themselves.

6.2 The Authority Control in the Group

The co-located-PP class had difficulty controlling authority and influencing one another directly so that they could perform the given tasks more easily and smoothly. On the other hand, the learners in the distributed-PP class could interrupt one another's access because two learners could input data at the same time. Moreover, the features of Nebraska could be supplied to a partner's work without any major obstacle. The learners in the distributed-PP class thus showed greater interest in cooperative activities, such as drawing together.

It was thus noted that the authority control function must be offered in the online-PP system so that a user will not be able to interrupt another user while the latter is working. Moreover, as the team members must have study periods with equal lengths, the timer function is also required so that the working times of the learners can be controlled.

6.3 The Pair Pressure

It was noted that the learners exerted pair pressure on one another while programming together. Such pair pressure is a psychological pressure that aims to reduce the number of learners' actions that have no correlation with learning [9]. In this study, the learners in the distributed-PP class demonstrated greater pair pressure compared to the learners in the co-located-PP class. As such, the learners in the distributed-PP class shared a remote interface, and no one among them

showed compulsive behavior. Therefore, it was observed that the learners in the distributed-PP class tried to learn collaboratively while those in the co-located-PP class tried to do their own work occasionally.

7 Conclusion

The researchers remain optimistic that PP can be effectively used in online-programming learning environments [9, 12]. However, we need educational trials of teaching and learning method for novices in divers environment because programming skill and algorithmic thinking ability are considered as basic literacy.

Therefore, this study experimented to conduct on the programming learning of novices with the use of Squeak eToys. Especially, we experimented on the divers learning effectiveness with comparing with online environment and offline one. The scores obtained by the following three classes were compared: the distributed-PP class, the co-located-PP class, and the individual-programming class.

As a result, it showed that learners could accomplish their programming tasks better both distributed and co-located PP than individual programming. The results showed that the distributed-pair method was not significantly different from the co-located-pair method. Furthermore, the pair programming method was shown to be more effective than the individual-programming method. The effects and feasibility of online pair programming were thus confirmed.

The results of this study will be a big help to computer programming teachers and will contribute to the dissemination of distance learning methods like e-learning [13], which need various individual learning methods, such as online PP. It can thus be said that this study is useful for the promotion of computer education.

Reference

[1] Mclver, L.: Syntactic and Semantic Issues in Introductory Programming Education, Monash University, Doctorial Thesis, 2001.

[2] Martin, J. L., Is Turing a better language for teaching programming than Pascal, Honors Dissertation, university of Stirling, Department of Computer Science, 1996.

[3] Wu, M., Luan, X., Leung, H. K. N., and Ying, J., Using System Dynamics Model to Simulate the Effect of Pair Programming in XP, WSEAS Transactions on Computer, Issue 10, Volume 6, October, 2007, pp.1111-1115.

[4] Cockburn, A. and Williams, L.: The Costs and Benefits of pair-programming. In Extreme Programming Examined, Addison Wesley Longman, 2001.

[5] Williams, L.: Strengthening the Case for Pair-Programming, IEEE Software. Submitted to IEEE Software, Volume 17, Issue 4, 2000, pp.19-25.

[6] Williams, L. and Kessler, R.: Pair-programming Illuminated, Boston, MA: Addison Wesley , 2002.

[7] Baheti, P. et. al.: Distributed Pair-programming : Empirical Studies and Supporting Environments, Technical Report, TR02-010, 2002.

[8] McDowell, C. et. al.: The Effects of Pair-Programming on Performance in an Introductory Programming Course, Presented at The 33rd ACM Technical Symposium on Computer Science Education, Feb., 2002.

[9] Shen, H. and Sun, C.: RECIPE: a prototype for Internet-based real-time collaborative programming, Proceedings of the 2nd Annual International Workshop on Collaborative Editing Systems in conjunction with ACM CSCW Conference, December 2 -6, Philadelphia, Pennsylvania, USA., 2000.

[10] Guzdial, M.: Mark Guzdial's Collaborative Software Laboratory, Swiki, Georgia Institute of Technology, URL at http://minnow.cc.gatech.edu/squeak/1356

[11] Williams, L. and Kessler R.: The Effects of “Pair-Pressure” and “Pair-Learning” on Software Engineering Education, Software Engineering Educator & Training 2000, Proceedings, 13th conference on march, 2000, pp.59-65.

[12] Jun, S., Kim, S., & Lee, W. Pair-Programming in Online Learning Environment, Information Processing Society of Japan (SSS’06), Joho Shori Gakkai Shinpojiumu Ronbunshu, 2006, 191-198

[13] Kim, S., Jun, S., and Han, S. Effective reasoning of learning styles using rough set theory in e-learning, The Proceeding of the 5th WSEAS International Conference on Education and Educational Technology (e-activities’06), 2006, pp.87-92.