Improving Learning of Programming Through E-Learning by Using Asynchronous Virtual Pair Programming

Abdullah Mohd ZIN
Department of Industrial Computing
Faculty of Information Science and Technology
National University of Malaysia
Bangi, MALAYSIA

Sufian IDRIS
Department of Computer Science
Faculty of Information Science and Technology
National University of Malaysia
Bangi, MALAYSIA

Nantha Kumar SUBRAMANIAM
Open University Malaysia (OUM)
Faculty of Information Technology and
Multimedia Communication
Kuala Lumpur, MALAYSIA

ABSTRACT

The problem of learning programming subjects, especially through distance learning and E-Learning, has been widely reported in literatures. Many attempts have been made to solve these problems. This has led to many new approaches in the techniques of learning of programming. One of the approaches that have been proposed is the use of virtual pair programming (VPP). Most of the studies about VPP in distance learning or e-learning environment focus on the use of the synchronous mode of collaboration between learners. Not much research have been done about asynchronous VPP. This paper describes how we have implemented VPP and a research that has been carried out to study the effectiveness of asynchronous VPP for learning of programming. In particular, this research study the effectiveness of asynchronous VPP in the learning of object-oriented programming among students at Open University Malaysia (OUM). The result of the research has shown that most of the learners have given positive feedback, indicating that they are happy with the use of asynchronous VPP. At the same time, learners did recommend some extra features that could be added in making asynchronous VPP more enjoyable.

Keywords: Pair-programming; Virtual Pair-programming; Object Oriented Programming

INTRODUCTION

Delivering program of Information Technology through distance learning or E-Learning is indeed a very challenging task. All programs in Information Technology include a few courses in programming which is considered to be the most important skill for IT professionals. Most of the other courses also require students to implement some programs in order to ensure that the students could be able to comprehend concepts that are presented in the course. However, a programming course is known to be a difficult course, both for the instructors and learners, even

in the normal mode of learning. Novice programmers suffer from a wide range of difficulties. According to Robins et. al. (2001), it will take about ten years of experience to turn a novice into an expert programmer.

Since 1970's, many innovative approaches have been proposed by educators to overcome problems in teaching and learning of programming. Most of these approaches can be classified under two main categories, namely: pedagogical and tools. Some examples of new pedagogical approaches in teaching of programming are "Tutorial-based teaching of introductory Programming" (Zachary 1994), "Methodology First and Language Second" (Zhu and Meng 2003) and "Process Model" approach (Gantenbein 1989). Some researchers have proposed that the programming curriculum need to be reshuffled to reflect learners need, while other proposed the imposition of mathematics to the learners who plan to take programming subjects. Another initiative was proposed by Carbone et. al. (2000) that encourages academics to consider the tasks they set for their learners since these tasks will affect the learning and understanding of programming, especially in a first year undergraduate course.

Number educational programming tools have been developed to assist learners in programming. One of them is BlueJ (Kölling and Rosenberg 2001), which provides an interactive environment for the teaching of Java that emphasizes visualization and experimentation. Other initiative in this area includes development of new languages that are "learner-friendly".

Another initiative that is slowly becoming more popular among many instructors is the concept of pair-programming. Pair Programming is one of the practices of eXtreme Programming (XP). XP is a development model for software that was first presented in 1996 by Ken Beck to take advantage of the weaknesses of the currently accepted methodologies (Crispin & House, 2003). The primary principles behind XP are based around communication, simplicity, feedback, and courage in developing software (William, 2004). In order to achieve this, XP has adopted 12 practices and pair programming is one of them. Pair programming is a practice in which two programmers work together at one computer collaborating on the same design, algorithm, code or test. While programming, the pair work side by side at a single workstation with one person designated as the 'driver' and the other person as 'observer' or 'navigator'. The driver has control over the keyboard and mouse and is responsible for entering programming code. The observer role is not passive; observers watch for potential defects and comment about programming approaches. These roles are switched as the programming session continues. Previous studies have shown that pair programmers produce higher quality code in essentially the same amount of time as solo programmers (Cockburn et. al., 2000).

All of the published studies to date indicate that the use of pair programming in the classroom has a positive impact on some aspect of learner performance or enjoyment, while none has demonstrated that the learning is compromised (McDowell et. al., 2003; Van De Grift, 2004). Initial experimentation with pair-learning also reveals benefits to computer science educators as well as in industry. Learners working in pairs are able to answer each others' questions. They no longer look to the teaching staff as their sole source of technical advice (McDowell et. al., 2003).

It seems that pair-programming is also the most suitable approach for learning of programming in the e-learning or in distance education environment for two important reasons. Firstly, learners in this environment are not in constant contact with their instructor. Secondly, learners in this environment normally have to work alone since they are not physically close to other learners. Pair programming

provide an environment for them to work with peers, which in turn will help to reduce anxiety and uncertainty of learners. In general, the reduction of anxiety and uncertainty tends to increase the learners' motivation and satisfaction with the learning process (Harasim et al., 1997). The instructor assumes the role of a cognitive and meta-cognitive coach rather than the knowledge holder and disseminator in the educational perspective of pair programming. The situation is fundamentally different from the traditional direct-transfer or one-way transmission model in which the instructor is the only source of knowledge or skills (Edelson et al., 1996). However, one of the key requirements of XP is strong and effective communication between the team members (Aiken 2004). To enable this strong level of communication among team members, XP emphasizes the need to have the team members physically located close to each other. This requirement is supported by a research which has indicated that pair programming is better than individual programming when the pairs are physically co-located (Baheti, 2002). In e-Learning or in distance education environment, learners are not physically co-located. The implementation of pair programming in this environment has led to the introduction of the concept of virtual pair programming or distributed pair programming (Baheti et. al., 2002; Hanks, 2004; Ho et. al. 2004; Kiercher et. al. 2001).

Studies that have been carried out suggested that virtual pair programming (VPP) can work and the output is comparable to the normal co-located pair programming (Nawrocki & Wojciechowski, 2001; Hanks, 2003). However, all of these studies focus on the synchronous mode of collaboration, for example by using tools such as NetMeeting, Instant Messaging and hypermedia-enhanced video streams. Since most of the learners in e-learning environment use dial-up connection to gain access to the Internet, synchronous mode of collaboration is not always practical. Hence, most of e-learning environment is based on asynchronous environment. In this environment learners communicate asynchronously by using e-mail and bulletin board. Asynchronous mode of collaboration is also appealing because they allow learners to manipulate time and space to their own advantage—learners can work when and where they please, without being constrained by the schedules or locations of others (Hiltz, 1998). Another reason for using asynchronous mode is that learners, while still operating on some shared set of data, context, information, or artefacts, do so largely independently of one another.

RESEARCH OBJECTIVES

The objective of this research is to investigate the effectiveness of using VPP in asynchronous mode of collaboration for learning of programming. In particular, we are looking at the effectiveness of using the approach in learning of Java Programming for e-learning learners at Open University Malaysia (OUM). Open University Malaysia was established in the year 2000. At the moment, the university has about 38,000 learners in five area of studies: information technology, business, science, engineering and education. The university employs three modes of studies: self-managed-learning through specially designed course material, face-to-face learning and web-based learning. For face-to-face learning, learners are required to attend 10 hours of tutorial sessions conducted by tutors at OUM learning centres, which are located throughout the country. For web-based learning, learners collaborate with other learners by using a specially developed system called LMS (Learner Management System).

A subject matter expert (SME) who is normally the lecturer at OUM is appointed to manage the course (for example to determine the course schedule, to set up assignments and preparing examination questions) as well to communicate with all of the tutors.

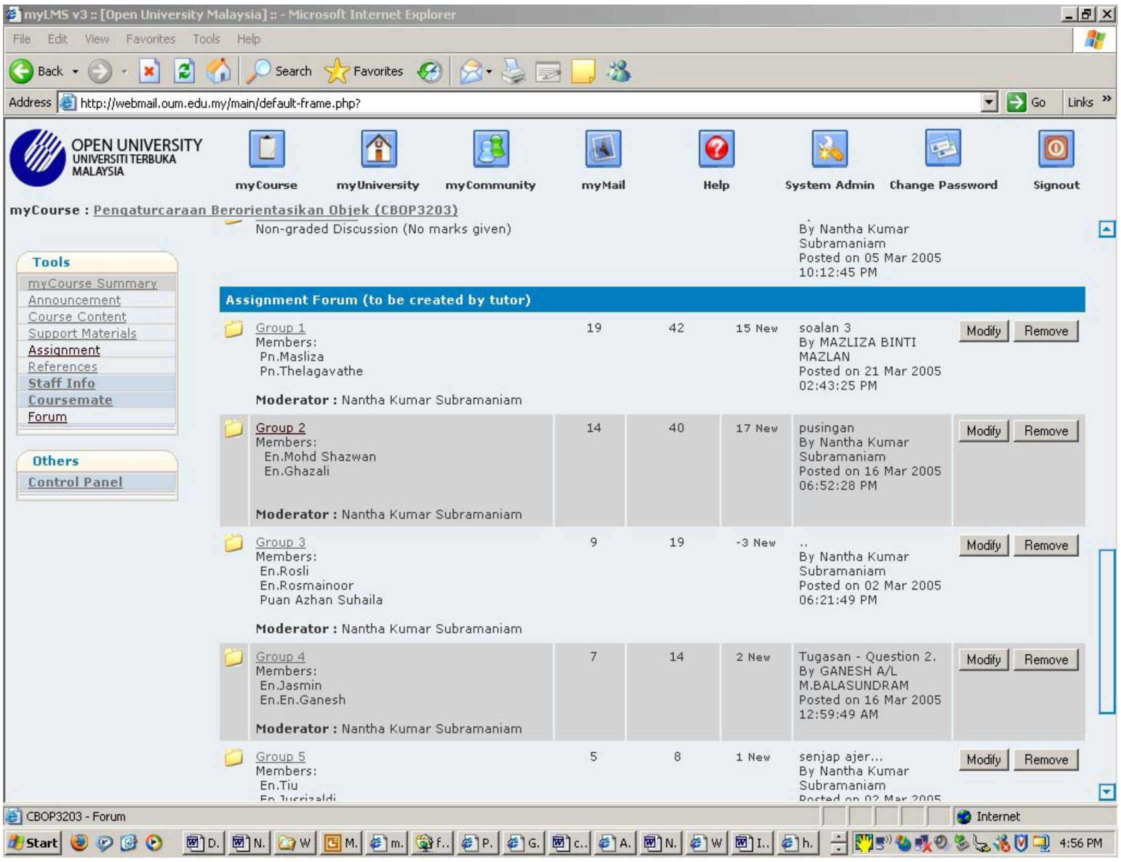
One of the compulsory courses for all learners majoring in information technology at OUM is object Oriented Programming. In this course, learners are required to write programs by using Java programming language. This is the third course in programming, since prior to this learners are required to take a course in C programming and Java Programming.

This object-oriented programming course covers topics on class, object and inheritance followed by the topics applet, event handling, exceptional handling and file manipulation. The basic syntax of Java including method, loops, conditionals, arrays, string have been covered in Java programming which was taught earlier.

RESEARCH METHODOLOGY

In this research, learners are required to use VPP in solving a programming project (later known as task) given to them. To introduce the concept of pair-programming, an overview about it has been attached on the programming assignment with clear instruction on what the tutor and the learner suppose to do.

The learner submits their programming solution (both hard and soft copy) and print-out of their discussion from the bulletin board on the fourth tutorial. In implementing VPP, a task consists of carefully designed problems that demand from the learner the acquisition of critical knowledge, problem-solving proficiencies and self-directed learning strategies.



myCourse : Pengaturcaraan Berorientasikan Objek (CBOP3203)

Non-graded Discussion (No marks given) — By Nantha Kumar Subramaniam Posted on 05 Mar 2005 10:12:45 PM

Assignment Forum (to be created by tutor)

Group 1 Members: Pn.Masliza Pn.Thelagavathe Moderator : Nantha Kumar Subramaniam	19	42	15 New	soalan 3 By MAZLIZA BINTI MAZLAN Posted on 21 Mar 2005 02:43:25 PM	Modify Remove
Group 2 Members: En.Mohd Shazwan En.Ghazali Moderator : Nantha Kumar Subramaniam	14	40	17 New	pusingan By Nantha Kumar Subramaniam Posted on 16 Mar 2005 06:52:28 PM	Modify Remove
Group 3 Members: En.Rosli En.Rosmainoor Puan Azhan Suhaila Moderator : Nantha Kumar Subramaniam	9	19	-3 New	.. By Nantha Kumar Subramaniam Posted on 02 Mar 2005 06:21:49 PM	Modify Remove
Group 4 Members: En.Jasmin En.En.Ganesh Moderator : Nantha Kumar Subramaniam	7	14	2 New	Tugasan - Question 2. By GANESH A/L M.BALASUNDRAM Posted on 16 Mar 2005 12:59:49 AM	Modify Remove
Group 5 Members: En.Tiu En.Jusrizaldi	5	8	1 New	senjap ajer... By Nantha Kumar Subramaniam Posted on 02 Mar 2005	Modify Remove

Figure: 1 Snapshot of LMS showing all the special pairs forum used to communicate among the members

The problem thus served as the organizing centre and the stimulus for learning and represented the vehicle that developed learners' creative and high-order thinking skills. There are two questions in this task:

- **Question 1 asks the learners to build a user interface using JApplet. This must use their creativity and innovation to make up with a impressive layout for this user interface**
- **Question 2 tests the learners on the concept of class and objects by asking them to develop a Java-based games application that will stimulate a dice.**

This assignment carries 20% of their final grade and their participation on the "pair forum" contributes 5% of their final grade. Each tutor is given a separate forum in LMS which can be used to communicate with their learners.

Tutor been asked to create subfolders in their forums for pair of the learners. A handout on how to create this forum had been prepared and distributed to the tutors. Each pair is given a different password to enter into their forum, in order to block learners from participate in forums that do not belong to them. The discussion in these forums will be monitored by the tutors, as shown in Figure: 1.

At the end of the course, a questionnaire about learners' perceptions on VPP was given. The questionnaire was specially designed to elicit the learners' perceptions toward the effect of the asynchronous collaborations that took place during the learning processes in VPP.

The main problem in designing the questionnaire is to determine the metrics that need to be measured. Most of the researches on capturing the learners' perception of pair programming are using different metrics.

Moreover, some of these metrics are more suitable for collocated pair programming environment, and thus could not be applied for VPP.

Some of the metrics that are used by researchers are as follows. Williams et. al. (2002) in investigating learners perception about pair programming get to look at their confidence level and motivation in learning computer programming, their perceived likelihood of success, likelihood of future use of programming skills, efficiency of the instructor and quality of the laboratory experience. McDowell et. al. (2003) has listed time, knowledge gained and productivity to investigate the impact of pair programming on learners' performance, perception and persistence.

Baheti et. al. (2002) in exploring pair programming in distributed object oriented teams has identified productivity and quality as main variables to get learners' feedback and at the same time take into the consideration about cooperation within team.

Van De Grift (2004) in investigating the learners' perception about pair programming has listed three variables, namely confidence; understand ability and efficiency in debugging. Hanks (2004) has used final exam performance confidence and gesturing as the variables and the same time also considered learners' experience with the tool.

Stotts et. al. (2003) has listed productivity, software quality and communication among the peers as the metrics. Based on metrics identified by other researchers, we have concluded that there are nine metrics to be used in this research. Since we are experimenting with e-learning learners, the quality of laboratory experience is not considered but replace by the effectiveness of the online learning tool used in collaborative programming which also identified by other researcher.

The learner performance in final exam is also not considered as we are investigating the learners' perceptions about VPP and not about the learners' performance.

Moreover, learners' higher perception about VPP would not necessarily lead to good performance in the final exam.

Table 1 shows items in the questionnaire. Questions 1-10 have been tagged with these metrics. Item 11 to Item 13 in the questionnaire are not tagged with any metrics as these questions are use to get some general feedback from the learner.

For questions 1-12, respondents are required to indicate their perception based on the scale of 1-4 (1– Strongly Agree, 2– Agree, 3– Do not Agree and 4– Strongly Do not Agree). For question 13, they are required to indicate either YES or NO.

Table: 1
Items in the questionnaire

No	Metrics	Statements
1	Confidence	Collaboration with my pair gives me more confidence in solving programming problems
2	Confidence	Collaboration with my pair gives me more confidence in writing Java programs
3	Tool	It easy to access forum in LMS
4	Tutor Role	Tutor roles is very much important in guiding the learner to obtain the output
5	Knowledge	I gained more new knowledge by participating in this pair forum
6	Learning Process	The discussion in the forum is more focus towards the problem that need to be solved
7	Learning Experience	Collaborating with my peer in solving the given task is a new rewarding experience
8	Peer	My group member gives concrete ideas
9	Persistence	This collaborative learning should be expanded to other subjects
10	Time	Less time is taken to obtain the solution using this approach
11	-	The collaboration in the forum will be more effective if it has more than TWO members
12	-	Pair collaboration with peer could replace the tutorial classes
13	-	The forum provided in LMS to collaborate with the peer is enough and no other facilities is required

RESULTS AND DISCUSSION

The result of the research is very encouraging. Learners showed special interest in participating in the discussion between their partners. We manage to collect 147 answers from 165 learners registered for the course.

The Cronbach's alpha reliability coefficient was calculated for the Item 1 – Item 11 and a high internal consistency for each of the dimensions was obtained which was 0.92. The analysis of the data involved extracting the means of each of the items with means of 2.50 representing the equilibrium point.

Means smaller than 2.50 reflected the degree of the respondents' agreement with the statement put forward while means with values more than 2.50 reflected the

degree of the respondents' disagreement with the statements put forward to them.The effects of the asynchronous collaborative programming process in an E-Learning environment are shown in Figure: 2. In general, the means for all statements (except question 12) are less than 2.

This result indicates that there is a high degree of agreement among the respondents towards statement put forward to them concerning the use of VPP.

It is evident that the learners perceived they had gained the confidence in programming by collaborating virtually with their peer ($\bar{x}$ =1.95) and this has contributed in more confidence on writing Java program ($\bar{x}$ =1.95). The positive result on this confidence could be resulted from new knowledge gained in participating in virtual pair-programming ($\bar{x}$ =2.06) and rewarding experience they gained in VPP ($\bar{x}$ =2.01).

The positive result on confidence they gained on programming made them support the use of virtual pair-programming in the future programming subject ($\bar{x}$ =2.01). The result also shows that the role of tutor is very important in guiding the learner in virtual pair-programming($\bar{x}$ =1.63).

The concrete ideas given by the peer ($\bar{x}$ =1.94) has made the discussion in the forum more focus on the problem that need to be solved ($\bar{x}$ =1.95).

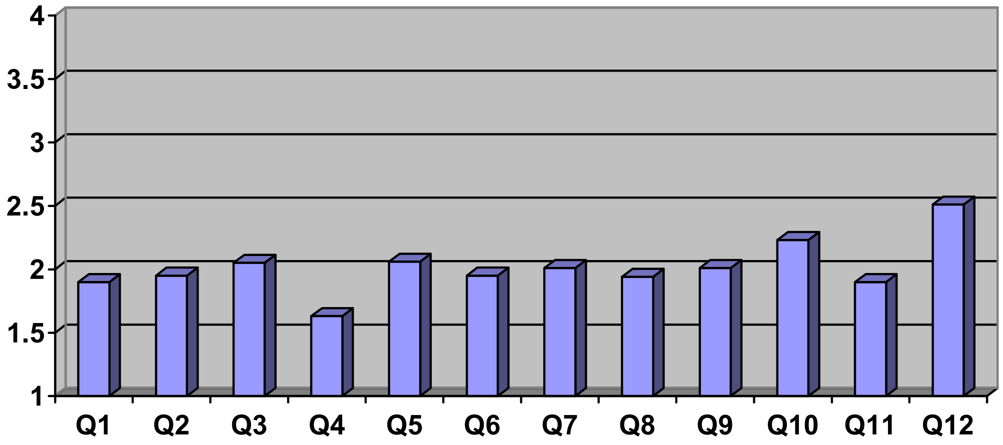


Figure: 2 The effects of asynchronous virtual pair Programming based on students' feedback

One interesting discovery is the result of item Number 11. For this item, most of the learners agree that having more than two members in a group would be more effective ($\bar{x}$ =1.90), although in general they are happy with their peer ($\bar{x}$ =1.94). In our opinion this is due to the fact that with only two learners in a group, there would be some delay in having the feedback from their partners. By having more than two, this delay could be minimized.

On questions Number 12, there is almost equal number of learners who agree and disagree that VPP could replace the physical face-to-face tutorial ($\bar{x}$ =2.51). This indicates that although generally learner happy with VPP but there are still a large percentage who would like to have face-to-face meetings with their tutors. This confirm to the opinion of Edwards et. al. (1997) who says that learning of programming subjects in e-learning institutions must have some face-to-face

interaction. Furthermore, in the Asian culture, it is a norm for learners to have regular face-to-face meetings with the instructors or tutors.

For question Number 13, which is an open-ended, 60% of the respondents do not agree that asynchronous mode of collocation is sufficient.

Those who do not agree mentioned that the some tools need to be added, such as online compiler, online notes and instant messaging tool (such as Yahoo Messenger). This result indicates that learners' wanted some kind of synchronous features in the forum.

CONCLUSION

This research was started with a single objective that is to investigate the effectiveness of using VPP in learning of programming. In particular, this research focus on the use of asynchronous mode of collaboration through the use of the university's learning management system.

More than one hundred learners have participated in this research. The result has revealed that the use of asynchronous VPP for learning of programming has produced many positive effects. Learners regarded VPP to be effective, motivating and enjoying. The research has also indicated that VPP has given them the confidence in programming.

This may be attributed to the new knowledge gained through the collaborative learning process between the learners as well between learners and tutors.

However, there are a few improvements that could be considered in order to make VPP more effective. Some learners have recommended that online compiler should be provided in the system.

At this particular moment, if two learners are discussing about a piece of program code, they may have to copy the program from the discussion forum and paste it to the stand-alone editor provided in their own personal computers before it can be compiled and run.

This arrangement can distract them from the focus of their discussion. By providing online compiler, the code can be compiled and run directly, and they can obtain the result immediately. Another recommendation from some of the learners is the inclusion of instant messaging.

Although the discussion forum provided by the learning management system has given them the ability to communicate, instant messaging is preferred since it can be used to give immediate feedback.

One interesting finding of this research is that most of the learners feel that the number of learners in a group should be more than two.

Although, two programmers working together is considered to be the best practice in software industry which adopted XP, it may not be the ideal number for learners in learning of programming.

In the next study, we are going to investigate this issue by varying the numbers of learners in a group and observe their performance. We are currently trying to correlate the learners activeness in VPP with their final examination result and also to study how learners' demographic profile, learning style and the way learners been paired influence the learners' perception about VPP.

BIODATA and CONTACT ADDRESSES of AUTHORS

Abdullah Mohd ZIN is a full professor at the Department of Industrial Computing, Faculty of Information Science and Technology, National University of Malaysia. He obtained his PhD (Computer Science) from Nottingham University (UK). He has published numerous articles in international peer-reviewed journals. His research interest includes collaborative learning tools, software management, E-Learning and foundation of programming. Currently he is the deputy dean as well as the head of programming research group at the faculty.

Abdullah Mohd ZIN
Department of Industrial Computing
Faculty of Information Science and Technology
National University of Malaysia
43600 Bangi, MALAYSIA
Email: amz@ftsm.ukm.my

Sufian IDRIS is a faculty member in the Computer Science Department of the Faculty of Information Science and Technology at National University of Malaysia (NUM). He holds a MSc in Computer Science from The University of Manchester, UK and a PhD in Computer Science from the University of Manchester Institute of Science and Technology, UK. He has been faculty member in NUM since 1988. His teaching interests are in the areas of Computer Programming, and Object-Oriented Analysis and Design.

His research interests and work are in the areas of Teaching of Programming, Programming Tools, Object-Oriented Development and Mobile Computing. He has published papers in journals such as the WSEAS Transactions on Computers and conferences such as the International Conference on Enterprise Information Systems.

Sufian IDRIS
Department of Computer Science
Faculty of Information Science and Technology
National University of Malaysia, 43600 Bangi, MALAYSIA
Email: sufian@ftsm.ukm.my

Nantha Kumar SUBRAMANIAM is a faculty member in the Faculty of Information Technology & Multimedia Communication, Open University Malaysia (OUM). He holds a Master in Computer Science from National University of Malaysia and currently a PhD (Computer Science) candidate in the same university. His research interest includes intelligent learning tools and object technology. His teaching interests are in the areas of Object-Oriented Programming, Object-Oriented Analysis & Design and Data Mining.

Nantha Kumar SUBRAMANIAM
Open University Malaysia (OUM)
Faculty of Information Technology & Multimedia Communication
50480 Kuala Lumpur, MALAYSIA
Email: nanthakumar@oum.edu.my

REFERENCES

AIKEN, J. (2004). Technical and Human Perspectives on Pair Programming. ACM SIGSOFT *Software Engineering Notes*. 29(5).

BAHETI, P., Gehringer, E. & Stotts, D. (2002) *Exploring the efficacy of distributed pair programming*. XP Universe 2002, Chicago.

BAHETI, P., Williams, L, Gehringer, E. and Stotts, D. (2002). Exploring Pair Programming in Distributed Object-Oriented Team Projects. Educator's Workshop, OOPSLA 2002, Seattle.

BAHETI, P., Laurie Williams, Edward Gehringer, David Stotts, and Jason Smith. (2002). *Distributed Pair Programming: Empirical Studies and Supporting Environments. Technical Report TR02-009*, Department of Computer Science, University of North Carolina at Chapel Hill.

CRISPIN, L., House, T. (2003). *Testing extreme programming*. Boston: Addison-Wesley.

HANKS, B. F. (2004). "Distributed pair programming: An empirical study". Extreme Programming and Agile Methods - XP/ Agile Universe 2004, Proceedings. 3134, pp. 81-91.

HANKS, B. F. (2003). Virtual Pair Programming. (2003). *Doctoral Symposium at the International Conference on Software Engineering (ICSE 2003),* Portland, OR.

CARBONE, A., John, H, Ian, M & Dick, G. (2000). Principles for Designing Programming Exercises to Minimize Poor Learning Behaviors in Students. Proceedings of the Australasian conference on Computing education.

COCKBURN, Alistair & Williams, Laurie. (2000) The Costs and Benefits of Pair Programming. eXtreme Programming and Flexible Processes in Software Engineering XP2000.

EDELSON, D. C., Pea, R.D. & Gomez, L. (1996) Constructivism in the collaboratory. In B G Wilson (Ed.). *Constructivist Learning Environment: Case Studies in Instructional Design*. Englewood Cliffs, New Jersey: Educational Technology Publication Inc, pp. 151-164.

EDWARDS, W.K., Elizabeth D. Mynatt, K. P., Mike J. S., Douglas B. T. & Marvin M. T. (1997). Designing and implementing asynchronous collaborative applications with Bayou. Proceedings of the 10th annual ACM symposium on User interface software and technology.

GANTENBEIN, R. E. (1989). Programming as Process: A Novel approach to teaching programming ACM SIGCSE. *Bulletin*, Proceedings of the twentieth SIGCSE technical symposium on Computer science education, 21(1).

HARASIM, L., Hiltz, S.R., Teles, L. & Turoff, M. (1997). *Learning Networks: A Field Guide to Teaching and Learning Online*. The MIT Press, Cambridge, MA.

HO, C., Raha, S., Gehringer, E, & Williams, L. (2004). Sangam: A Distributed Pair Programming Plug-in for Eclipse , Eclipse Technology Exchange (Workshop) at the Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA).

HILTZ, S. R. (1998). Collaborative Learning in Asynchronous Learning Networks: Building Learning Communities. Invited Address at "WEB98" Orlando Florida.

KIERCHER, M., P. Jain, A. Corsaro & D. Levine. (2001). *Distributed eXtreme Programming Appeared at XP2001.*

KÖLLING, M. & John Rosenberg. (2001). Guidelines for teaching object orientation with Java. ACM SIGCSE. *Bulletin*, Proceedings of the 6th annual conference on Innovation and technology in computer science education, 33(3).

McDOWELL, C., Brian, H. & Linda, W. (2003). Experimenting with pair programming in the classroom. ACM SIGCSE. *Bulletin*, Proceedings of the 6th annual conference on Innovation and technology in computer science education, 33(3).

McDOWELL, C., Linda W., Bullock, H.E. & Fernald, J. (2003). The Impact of Pair Programming on Student Performance, Perception and Persistence. Proceedings of the 25th International Conference on Software Engineering (ICSE'03).

NAWROCKI, J. & Wojciechowski, A. (2001) A. Experimental Evaluation of Pair Programming. Proceedings of the 12th European Software Control and Metrics Conference, ESCOM 2001), 269-276.

ROBINS. A, Janet, R. & Nathan, R. (2001). My program is correct but it doesn't run: A review of novice programming and a study of an introductory programming paper Department of Computer Science, University of Otago. Technical Report.

STOTTS, D., Williams, L, Nagappan, N., Baheti, P., Jen, D. & Jackson, A. (2003). Virtual Teaming: Experiments and Experiences in Distributed Pair Programming, XP/Agile Universe.

VAN DE GRIFT, T. (2004). Coupling pair programming and writing: learning about students' perceptions and processes. ACM SIGCSE. *Bulletin*, Proceedings of the 6th annual conference on Innovation and technology in computer science education, 33(3).

VYGOTSKY, L. S. (1978). *Mind in Society: The Development of Higher Psychological Processes*, M. Cole, V. John-Steiner, S. Scribner, and E. Souberman, Eds. Cambridge, MA: Harvard University Press.

WILLIAM, W. (2004). XP123 - exploring extreme programming. Retrieved 22 May 2006 from http://www.xp123.com/xplor/xp0502/index.shtml

WILLIAMS, L., Eric W., Kai Y., Miriam F. & Carol M. (2002). In Support of Pair Programming in the Introductory Computer Science Course. Computer Science Education.

ZACHARY, J. L. (1994). Tutorial-based teaching introductory programming classes. ACM IGCSE *Bulletin*, Proceedings of the twenty-fifth SIGCSE symposium on Computer science education, 26(1).

ZHU, H. & Meng, C. Z. (2003). Methodology First and Language Second: A Way to Teach Object-Oriented Programming. Companion of the 18th annual ACM SIGPLAN conference on Object-oriented programming, systems, languages and applications.