Developing Problem-Solving Skills and Pair Programming Strategy for a Fundamental Computer Programming Course

Siriporn Mikum, King Mongkut's University of Technology Thonburi, Thailand
Surachai Suksakulchai, Sripatum University Chonburi Campus, Thailand
Settachai Chaisanit, Sripatum University Chonburi Campus, Thailand

The Asian Conference on Education 2014
Official Conference Proceedings

Abstract
Many studies have shown that when learning programming students don't have basic problem solving skills and don't know how to create algorithms. Our overall Objective of this study was to help students develop the foundational capabilities needed to become successful programmers and to help students learn effective programming skills. The main focus of this paper is a literature review research relating to problem-solving skills that help students practice more efficient analysis, planning and design skills for the development of programs. A problem-solving activity consists of five processes: analysis to understand; development the plans, carrying out the plans, evaluation the plans and reflection on the problem-solving process. These activities have used the method of pair programming activities to enhance teamwork and communication during learning. The processes of activities were consistent with program learning. This research was experimental pilot for the quality of learning framework, which the results show that, the students improved programming skills and the satisfaction of learner on high level in process.

Keywords: Problem-solving skills, Pair programming, Fundamental Computer programming

Introduction

Programming is a basic course for students in the field of computer and Information Technology. Learning to computer program is a difficult process for many students. The difficulty of the process causes a high failure rate in many schools. Several authors have discussed different reasons for such problems (Sloane, 1988; Gomes, 1998; Soloway, 1989; Jenkins, 2002; Lahtinen, 2005). A study by Mikum (2013) and Gomes (2007) found that students don't have basic problem solving skills and don't know how to create algorithms. Developing problem solving skills should be a priority.

There are many studies that show a positive effect between computer programming problems solving ability (Battista, 1986; Kurshan, 1985). Problem solving is essential to computer programming and requires multiple abilities that students often don't have (Gomes, 2007). And according to Craig further research has suggested teaching lab between teachers and students should have learned in the problem-solving process. The process of developing problem solving skills helps to improve the students' ability to understand and plan for problems.

However, another important part of learning to program is the happiness of the students while learning. Many researches have reported how enjoyment of learning programming will increase the learning environment in the class (Mikum, 2014). Pair programming, which it has helped create an environment for learning programming better and including of benefits that it help to programs of higher quality, helped improve retention, understanding of the programming process, and other (Laurie Williams, 2010; Mikum,2014).

However, the developing of programming skills to strive to improve the quality of the students should have developed to be continuously. The purpose of this paper is to propose a practical strategy of Problem solving activities and pair programming for improving the fundamental computer programming course.

Background Principles

Pair programming

Pair programming involves a type of learning in which two programmers combine efforts on the same problem, algorithm, design, code or test, and work together at the same computer (Williams, 2010; Anderson, 2012; Cockburn,2000; Ma,2004). One student, the driver, is assigned to design, write the code and test programs. The other student, called the navigator, watches to see if the driver makes mistakes and then gives advice to help fix the mistakes. The communication between the driver and navigator is important. And it is a good idea to switch roles between the drivers becomes the navigator (Williams, 2002).

When teachers choose to use the pair programming technique on learning programming course, learners will benefit both, in terms of, the development the programming process and their feelings about their studies. For instance, Students get higher quality programming when pair programming is used. The work in pairs planning, sharing ideas and combining solutions increases the cognitive knowledge of the of programming process, as well as helps students learn programming better, get

happy feelings about learning in general and increases students confidence in their programming skills (Han, 2010).

Choosing the right partner in pair programming is essential to success. According to Grigori Melnik (2002) and Dean Sanders (2002), the skill levels of the learners are necessary to achievement in their program. Lynda Thomas (2003) found that choosing a good pair programming when a partner is capable of the same skill levels. Or the choosing a pair programming of students is the job of students chooses their partner (Mikum, 2014). When they completed programs, they get switch roles for this will spread information and knowledge will be spread throughout their pair (Williams, 2003).

Problem-solving skills

Learning programming is based on development of the student's problem-solving ability (Gomes, 2007; Vanlengen, 1990). The authors define the "problem" using a definition from Gil Pérez et al. (1988) who consider the problem as a situation where the solution is not clear. Perales (1993) considers the problem as any situation that produces, on one hand, level of uncertainty, and on the other, an expression in search of a solution. According to Mayer (1998) it is a complex concept containing cognitive, metacognitive and motivational aspects.

A literature review of the problem-solving skills encourages us to find ways to improve this subject in terms of different knowledge domains.

For instance, Polya (1957), in his book "How to Solve It", describes ideas on how to enhance the student's problem-solving skills, which he thinks involve four-phases, namely: 1) understand the problem; 2) devise a plan - it often means looking at related or simpler problems; 3) carry out the plan and 4) look back.

Bransford and Stein (1984) presented the IDEAL model, and using the following steps: 1) Identification of the problem; 2) Definition of the problem with precision; 3) Exploration of strategies to reach the problem solution (based in previous knowledge and experiences); 4) Action, in the sense of the execution of the previously planned; 5) Learn (or Look back) relative to the observation of the effect of the carried through actions and learning according to the evaluation of the results of these actions.

Sternberg and Davidson (1989) suggested other steps: 1) problem identification; 2) selection of the mental operation to solve it with success; 3) internal and external representation of the information, in a clear way; 4) selection of an adequate strategy; 5) distribution of the available resources; vi) monitor the different moments of problem solving.

Pretz and Colleagues have divided the problem-solving process in yet other stages: 1) to recognize or to identify the problem; 2) to define and to represent the problem mentally; 3) to develop a resolution strategy; 4) to organize the knowledge concerning the problem; 5) to attribute mental and physical resources to solve the problem; 6) to monitor ideas so not to divert from the main goal; 7) to evaluate and correct the solution.

The University of Washington (2003) provides details that problem solving is an iterative, or cyclical process and describes the various steps to solve problem, namely: 1) Identify the problem; 2) Define the problem; 3) Collect, evaluate and organize information about the problem(Determine what information will be relevant, classify and categorize relevant information) ;4) Create or select a strategy to resolve the problem ;5) Allocate resources to solve the problem(Encourage students to develop timelines, action plans, progress reports); 6) Monitor the problem solving process (Ask students to submit regular progress reports or updates to ensure deadlines are met); 7) evaluate the final solution (to evaluate their final solution about e.g., an accountant; a manager; a researcher).

Also Santucci offers synthetic forms of abbreviations FARE techniques and methods based on the original model of Polya, referring to the following stages: 1) Focusing on the creation, selection and definition of the problem, deciding and what is necessary to know; 2) Analyzing, by collecting reference data, determining the relevant factors, and generating alternative solutions (or action plans); 3) Resolving, by selecting one solution, developing a plan for update and persistence in the organization to reach the awaited result; 4) Execution, finding a solution, controlling the impact during the plan implementation (evaluation of the results).

Almeida (2004) describes the model of the problem-solving in the following five steps. 1) Recognition, definition and identification of the problem; 2) Analysis of the problem and generation of alternative solutions; 3) Development of plans; evaluation of the alternatives and selection of one of them; 4) Selection and effective implementation of the alternative solutions; 5) Evaluation and follow-up or solution testing.

Anabela Gomes and António José Mendes (2007) describe in their work how to solve programming problems. It includes the following phases: 1) Understanding the problem (to define the problem and understand the aspects that are not clear); 2) Characterizing the problem (Looking for a related or similar problems that students had solved the problem); 3) Representing the problem (Students represent problems and answer questions to demonstrate understanding more.); 4) Solving the problem (making decisions, designing a system to meet certain goals, diagnosing and proposing a solution); 5) Reflecting on the solution(examining solutions and looking for information or clarification; evaluation of solutions from different perspectives.); 6) Communicating the problem solution(to communicate solutions can help students to examine problems that were previously not understood, and can also reflect the solution on production).

For a better understanding of the solution process it would be useful to combine the various methods of synthesis from many authors. This study used the following five processes based on the methods of other researchers. This study is focused on using this synthesis in relation to computer programming. The guidelines of problem solving of many researchers are consistent with the practice of programming. They include the following processes: 1) analysis to understand; 2) development the plans; 3) carrying out the plan; 4) evaluation of the plan; 5) reflection on the problem-solving process.

Instruction Design
Problems on learning computer programming fundamentals are the students lack the problem solving skills and cannot to design algorithms (Mikum, 2013; Gomes, 2007). Be said that, when the students practices a problem-solving skills, that this skills will help to develop better programs. The steps of practice skills, which are consistent with the steps of programming and the pair programming determine a role with the problem-solving skills and pair programming expected to support their learning and understanding.

This study hypothesizes that if students practice the above steps on program learning, the steps will lead to the development of better programmers as well as increasing the quality of their programming.

Processes of Problem-solving activities
Problem-solving activities consists of five processes, namely analysis to understand, development the plans, carrying out the plan, evaluation of the plan and reflection on the problem-solving process, Figure1.

The pre-activities, the learner choosing a partner had conditions of pairing by divided grade point average are three levels, namely high, medium, low respectively (Wang, 2012), then the instructor assigns a problem to the pair learners and describes the role of learner and how to learn in the assignment. The pair programming means two programmer works together, one student is the driver and is responsible to design, write the code and test. The other one, called the navigator, is responsible to observe the work of the driver looking for mistakes and providing strategic suggestions.
Table 1: The roles in the problem-solving activities

Driver	A student who analysis, designs, writes the code and tests the assigned programs
Navigator	A student who watches to see if the driver makes mistakes and then gives advice to help fix the mistakes
Teacher	The teacher of a programming course, in charge of giving assignment, observation and brief in assignment and summarize assignments

The first processes, in analysis to understand, students get assignments from teacher and then the driver analyzes the problem programming on Input-Process-Output analysis (I-P-O Analysis). This encourages a more clear analysis by the students. This is a process to analyze and understand the assignment and can be divided into three items namely: input, process, and output.

In the input item, the driver analyses data and the values, defines variables and types of variables in the assignment.

In the process item, the driver plans an approach to problem solving and chooses the command or function to be used in programming.

In the output item, the driver designs the result of programming for a guide for the programming.

The entire process of analysis is issued by driver under observation of the navigator.

The second processes, in the development of the plans, the driver designs the programming on process planning, the flowchart design uses the swim lane technique, a symbol used in process flow diagrams, or flowcharts, to see differently the responsibilities for sub-processes. The driver describes the design of programs to the navigator before the driver writes the program. This processes helps guide the problem solving of programming.

The third processes, carrying out the plan, the pairs analyze the problem and design of completed programs. They start to code the program as designed by driver under the observation of the navigator.

The fourth process, evaluation of the plan, tests and debugs the programs based on the errors found. Students work together and brainstorm the problem solving of the driver and navigator.

In The fifth process, reflection on the problem-solving process, students discuss and comment on the program of friends presented in the classroom. Then share questions and answers to better understood the programming.

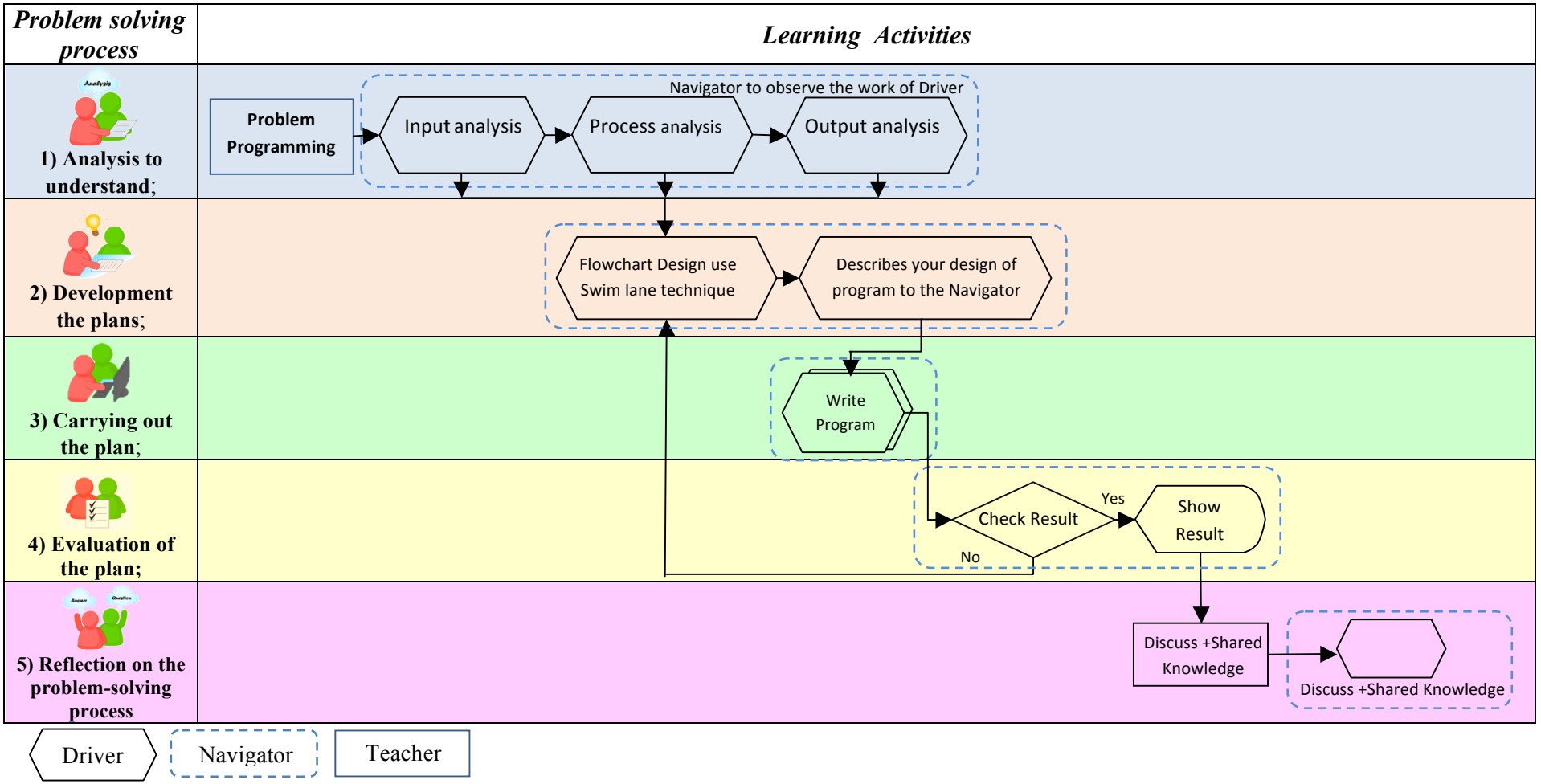

Figure1. **P**roblem solving Activities and **P**air Programming for improving the fundamental Computer **P**rogramming Framework (**P^3 framework**)

Findings

Experimental Pilot

Literature reviews on various topics are basic knowledge of the concept for developing computer programming skills (Figure1.); the concept is practice for problem-solving skills, problem analysis, and design of program. And this concept was tested with the 10 freshman students in area of major Education Information Technology, Rajamangala University of Technology Thanyaburi. The result found that:

Table2. : Table shows the mean score according the test with pre-test and post-test

Test	$\bar{X}$	SD.	$\bar{D}$	$SD_{\bar{D}}$	t	Sig.
Pre-test	4.9000	1.37032	2.80000	1.98886	4.452	.002*
Post-test	7.7000	1.88856				

Table2. Shows the mean score in the test of students, post -test is higher than pre- test, and the difference was statistically significant ($p < .05$). The students improve programming skills on P^3 framework.

Satisfaction of learner

This study explored the satisfaction of learning activity with the experimental pilot group on Figure 2.

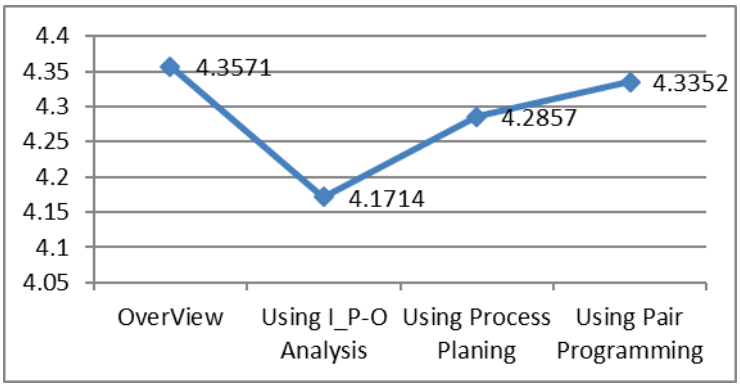

Figure2. Average of satisfaction

In satisfaction queries, there are four aspects. The students provided the average of satisfaction on high level in all aspects. An interesting finding is that on the high level, the students need to learn other subjects like the activity in subject and the students have opportunity to share knowledge and show ideas with their partners. The scores show that the students are satisfied with the learning activity.

Interview

This study involved interviewing the students. Students who participated in the interview said that the process of learning activity and working together helped them understand the problems of programming and designing better programs with flowcharts, it helped to have a guide to start their thinking process. Students also said that working together was a happy way to help with programming. They noted that they made friends who provided suggestions, shared knowledge and solved problems with their as a pair. Additionally, the pair work was more satisfying and increased the happiness of the developers.

Discussion

Various researches have the aim to develop programming skills of novice programmers. This current research aims are the same. Pair programming is a style of programming in which two programmers work together on the same process of

problem solving, which focuses on analysis to understand the problem and design programs. The learning framework helped them understand the problems of programming and better program design with flowcharts. Additionally, this process improves the quality of the analysis of problems, quality of the designing of programs and quality of code. It also was a positive experience for the students.

In this problems-solving process, the I-P-O Analysis encourages the ability to analyze problems better such as analysis of data, variable, and planning solutions. In addition, the process planning encourages clear plans in the designing of programs and increased the quality of code.

The switching role encourages students to practice communication, reasoning thinking, and sharing knowledge and encourages learning by doing and improves programming skills. And in areas, analysis problems and observation errors in programming occurred among participants who received the navigator role (Han, 2010).

The pairing with students of high-level skills that same level for programming together found that the pairing of the best and the student low-level skills for pair work together every process was slow as well. The pairs shared ideas and accepted criticism and the students were enthusiastic and willing to work in pairs. This also enhanced communication skills for the pairs (Wang, 2012).

Conclusion

Learning to computer program is a hard process for many students. The difficult process causes a high failure rate in many schools. This paper presents a learning framework using a problem-solving activity based on pair programming for improving programming skills.

When the learning framework was used in programming courses, it was found to have a positive effect on learning. The results showed the mean score in the test of students, post-test was higher than pre-test, see Table1. This table shows how the learning framework encouraged learning programming. The enjoyment of developers also increased.

Future work will be explore the learning framework in the area of programming to examine whether the development of programming skills leads to even better performance.

Acknowledgements

This work was funded by Rajamangala University of Technology Thanyaburi, Thailand.

Siriporn Mikum. Ph.D. Candidate (Learning Innovation and Technology) Faculty of Industrial Education and Technology, King Mongkut's University of Technology Thonburi, Bangkok 10140 Thailand (corresponding author to provide phone: +66-2470-8508; fax: +66-2427-8886; e-mail: 54501808@st.kmutt.ac.th).

Surachai Suksakulchai. Author is with the Department of Electrical Technology Education, Faculty of Industrial Education and Technology, King Mongkut's University of Technology Thonburi, Bangkok 10140 Thailand (e-mail: surachai.suk@kmutt.ac.th).

Settachai Chaisanit. Author is with School of Information Technology, Sripatum University Chonburi Campus, Chonburi 20000 Thailand (e-mail: settachai.ch@gmail.com).

Reference

Sloane, K. D. & Linn, M. C. (1988). Instructional Conditions in Pascal Programming

Classes. R.E. Mayer (Ed.), Teaching and learning computer programming. Hillsdale, New Jersey: Lawrence Erlbaum Associates, 137-152.
Gomes, A. & Mendes, A. J. (1998). Ambiente de suporte à aprendizagem de conceitos básicos de programação. In Actas do 3º Simpósio de Investigação e Desenvolvimento de Software Educativo, Évora, Portugal, Setembro.
Soloway, E., & J. Spohrer. (1989). Studying the Novice Programmer. Lawrence Erlbaum Associates, Hillsdale, New Jersey.
Jenkins, T. (2002). On the difficulty of learning to program. In *Proc. of the 3rd Annual LTSN_ICS Conference*, Loughborough University, United Kingdom, August 27-29, The Higher Education Academy, 53-58.
Lahtinen, E., Ala-Mutka, K. & Järvinen, H-M. (2005). A study of difficulties of novice programmers. In *Proc of the 10th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education*, Monte de Caparica, Portugal, June 27-29, 14-18.
Williams, L. (2010). Pair Programming. Encyclopedia of Software Engineering, 2.
Williams, L., Kessler R. R., Cunningham, W., & Jeffries, R. (2000). Strengthening the Case for Pair- Programming. *IEEE Software*, 17(4): 19-25.
Anderson, N., & Gegg-Harrison, T. (2012). Pair^2 Learning = Pair Programming x Pair Teaching. In *Proceeding WCCCE '12 Proceedings of the Seventeenth Western Canadian Conference on Computing Education,* Vancouver, British Columbia, Canada: Association for Computing Machinery, 2-6.
Cockburn, A., & Williams, L. (2000). The Costs and Benefits of Pair Programming. *eXtreme Programming and Flexible Processes in Software Engineering XP2000,* 223-247.
Almeida, A. C.(2004). Cognição como Resolução de Problemas: Novos horizontes para a investigação e intervenção em Psicologia e Educação. PhD Thesis. Faculdade de Psicologia e Ciências da Educação da Universidade de Coimbra: Portuguese.
Gomes, A. & Mendes, A. J. (2007). Problem solving in programming. *PPIG'07 Work in Progress Report*, http://www.ppig.org/workshops/19th-programme.html {Accessed January 2014}, 216-228.
Estivill-Castro, V. (2010). Concrete programming for problem solving skills. *2nd International Conference on Education and New Learning Technologies (EDULEARN10 Proceedings)*, Barcelona, Spain, 4189-4197.
Vanlengen, C. A. & Maddux, C. D.(1990). does instruction in computer programming improve problem-solving ability. *Journal of IS Education* ,CIS Educator Forum, 2(2): 11-16.
Ma, L., Ferguson, J., Roper, M., Wilson, J., & Wood, M. (2004). A Collaborative Approach to Learning Programming: A Hybrid Learning Model. In: *6th Annual Higher Education Academy Subject Network for Information Computer Science conference,* Academy for Information and Computer Sciences, York.
Battista, M. T., & Battista, D. H. C. (1986). The effects of Logo and CAI problem-solving environments on problem-solving abilities and mathematics achievement. *Computers in Human Behavior* ,2(3): 183–193.
Mikum, S., Baden, J., Suksakulchai, S., & Chaisanit, S. (2014). Pair programming as a tool in programming courses' strategies for Thai students. *In Education and Leadership in Glocalization : What does "think globally, act locally" mean for education around the world?*, Phuket, Thailand, 254-259.
Kurshan, B., & Williams, J. (1985). The effect of the computer on problem solving

skills, ERIC, 17.
Mayer, R. E.(1998). Cognitive, metacognitive and motivational aspects of problem solving. *Instructional Science*, 26(1-2): 49-63.
Cross, Indianapolis, Hackett Pub. Co. 1993. Polya, G. *How to Solve It*. Princeton University Press, Princeton, New Jersey, (1945).
University of Washington, Teaching and Learning. (2003*). Common Teaching Situations: Teaching Problem Solving.* (Retrieved from the World Wide Web on 1 September, 2004),http://depts.washington.edu/cidrweb/TAHandbook/ProblemSolving.html
Bransford, J. D. & Stein, B. S. (1984). *The IDEAL problem solver: A guide for improving thinking, learning and creativity*. New York: W. H. Freeman and Company.
Sternberg, R. J. & Davidson, J.E. (1989). A four-prong model for intellectual skills development. *Journal of Research and Development in Education*, 22(3): 22-28.
Santucci, U. Problem setting. Available at http://web.tiscaline.it/problemsetting/
Pretz, J. E., Naples, A. J. & Sternberg, R. J. (2003). Recognizing, defining and representing problems. J. Davidson and R. Sternberg. (Ed.). *The psychology of problem solving*. New York: Cambridge University Press.

Melnik, G. & Maurer, F. (2002). "Perceptions of Agile Practices: A Student Survey," *Extreme Programming and Agile Methods — XP/Agile Universe 2002,* Chicago, IL, 241-250.
Sanders, D.(2002). "Student Perceptions of the Suitability of Extreme and Pair Programming," in *Extreme Programming Examined*, M. Marchesi, G. Succi, D. Wells, and L. Williams, Eds.Boston, MA: Addison Wesley, 2002, 261-271.
Thomas, L., Ratcliffe, M. & Robertson, A. (2003)."Code Warriors and Code-a-Phobes: A study in attitude and pair programming," presented at SIGCSE, Reno, NV.
Han, K., Lee, E.& Lee, Y. (2010). The Impact of a Peer-Learning Agent Based on Pair Programming in a Programming Course, IEEE TRANSACTIONS ON EDUCATION, 53(2), 318-327.
Wang, Y., Li, H., Feng, Y., Jiang, Y. & Liu Y. (2012). Assessment of programming language learning based on peer code review model: Implementation and experience report. *Computers & Education* 59: 412–422.