S. L. Wong et al. (Eds.) (2010). *Proceedings of the 18th International Conference on Computers in Education*. Putrajaya, Malaysia: Asia-Pacific Society for Computers in Education.

The Effect of Scaffolding Support on Programming Performance and the Use of Self-regulation in Learning Computer Programming

Hsiao-lan LEE, Han, Yi-TI & Ming-Puu CHEN
Graduate Institute of Information & Computer Education, National Taiwan Normal University, Taiwan
supersharon@gmail.com

Abstract: The present study is to explore the students' use of self-regulation and their learning performance of computer programming under the framework of Experiential-based Learning of Computer Science (ELCS) in learning computer programming in an elementary school. The problem-solving support (PS group) and the procedural support (PR group) are regarded as two different scaffolding supports to guide learners to learn computer programming concepts. The result shows that the PS group has better performance in both advance and entire concept learning of computer programming. And the PR group asks more help to assist themselves than the PS group.

Keywords: self-regulation, computer programming, problem-solving, experiential learning

Introduction

Computer science is regarded as an advanced and significant ability for learners to adapt to today's digital and information world, but the knowledge and skills of it lack of the link to learners' life experience and prior knowledge [2, 23]. To bridge the gap between the abstract knowledge of computer science and learners' concrete life experience, the instructional framework of Experiential-based Learning of Computer Science (ELCS) based on the experiential learning perspective may support learners to learn abstract concepts of computer science through learning activities employing examples of concrete experiences, contexts for reflective observation, conceptual models for abstract conceptualization and contexts for active experimentation [2].

Moreover, computer science in school education should focus mainly on the learning of conceptual, strategic and even problem-solving knowledge and skills, and it also means to emphasize the valuable learning activities of logic, design, problem-solving, critical reflection, and self-expression [3, 24]. And it is critical for learners to acquire a coherent and broad understanding of principles, methodologies and applications of computer science and develop the computer science skills of algorithm development, problem-solving, and programming [22]. And especially, programming language instruction has been shown to enhance a variety of specific problem-solving skills [15]. Thus, computer programming learning should be that learning to program may have benefits from teaching general-purpose problem-solving and thinking skills and may help learners appreciate and understand how computers work [20].

In the process of learning programming problem-solving, training and experience in the metacognitive skills may increase learners' problem-solving ability [15]. And self-regulation is a learning cycle of metacognition in learners' learning process. It focuses attention on how learners personally activate, alter and sustain their learning practices in

specific contexts and includes self-monitoring of one's activities, applying personal standards for judging and directing one's performances, enlisting self-reactive influences to guide and motivate one's efforts, and employing appropriate strategies to achieve success [25, 26, 28]. So due to the support of the framework of ELCS, learners may apply the self-regulation skills to enhance the programming problem-solving learning.
Thus, in the present study, the learners were provided an instruction to learn computer programming through two different scaffolding supports, one of which was based on the framework of ELCS. And the learners' learning performance of computer programming and their use of self-regulation would be the important factors to investigate.

1. Literature review

1.1 Learning Computer Programming problem-solving

Computer programming involves the design and development of problem-solving algorithms and is treated as a valuable medium to develop problem-solving skills [3, 15]. However, while computer programming is considered as a valuable learning activity, it is also a complex and difficult task for novices to master [10]. Due to computer programming regarded as a problem-solving activity, it emphasizes that learning programming should be placed on the problem to be solved and the steps required for the solution [11, 13]. And computer programming mainly consists of three activities: problem identification and analysis, programming to tackle problems, and program representation in a computer coded language [21]. So an adaptable instructional framework could be treated as a learning support for learners and is necessary to use in learning computer programming [15].
However, computer science concepts contain abstract concept so that many learners are unable to achieve these learning goals because they are left with a fragile knowledge of programming [12, 23, 24]. And on the other hand, because of the absence of the link to learners' life experience and prior knowledge, their motivation and enthusiasm are diminished [24]. Thus, the design and development of learning support based on ELCS in learning programming problem-solving could be the challenging, and problem-bounded design activity includes abstract cognitive activities and involves complex cognitive processes as it requires the solving of continuous problems during execution [2, 21].

1.2 Self-regulation in ELCS

Self-regulation emphasizes how learners personally activate, alter, and sustain their learning practices in specific contexts, and even high-ability learners often do not achieve ideally because of their failure to use or control contextually specific cognitive, affective, and behavioral learning processes [4, 26]. It also means that learning by using strategies and goals, regulating and monitoring certain aspects of cognition, behavior, and motivation, and modifying behavior to achieve a desired goal [16]. Thus, while proceeding the design and development of computer programming regarded as the complex and abstract problem-solving process, learners may be conducted into the subprocesses in self-regulation: forethought, performance, and self-reflection, which stress on the learners' metacognitive control in the whole process of learning [14, 27].
However, learners are rarely given choices regarding academic tasks to pursue, methods for carrying out complex assignments, or study partners, and few teachers tend to encourage learners to establish specific goals for their academic work or teach explicit study strategies [27]. Also, learners are hardly asked to self-evaluate their work or estimate their competence on new tasks. But in learning computer programming, metacognitive skills is

critical for learners to become aware of the appropriate and effective strategies that are needed to solve a variety of problems [5, 15].
Furthermore, the framework of ELCS shown in Figure 1 is based on the experiential learning theory, a four-stage learning cycle, including abilities-concrete experience, reflective observation, abstract conceptualization, and active experimentation [6]. It also mainly focuses on providing flexible-to-adopt, ease-of-use, from-concrete-to-abstract, and from-observation-to-experimentation digital instructional materials to strengthen technology infusion in high school classrooms [2].

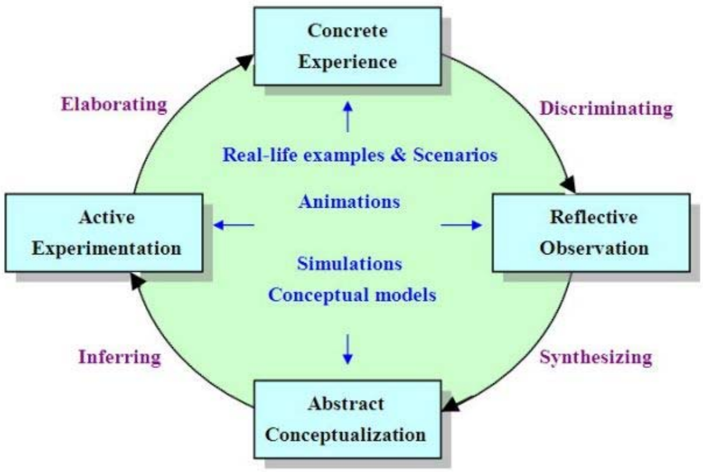

Fig. 1 The framework of Experiential-based Learning of Computer Science

As a result, the purpose of the study is to investigate learners' learning performance of computer programming and their use of self-regulation under the scaffolding support of framework of ELCS while learners face of difficulties, stressors, or competing attractions in learning computer programming.

2. Methodology

2.1 Participants

Seventy-seven sixth-grade students (37 males and 40 females) from north Taiwan participated in a five-week computer programming learning activity in this study. And the students were assigned to two conditions: Problem-solving Support Group (PS group; n = 38; 18 males and 20 females) and Procedural Support Group (PR group; n = 39; 19 males and 20 females). The domain knowledge for the participants was computer programming learning in Stagecast Creator. And the participants had little prior knowledge of the topics for computer programming.

2.2 Research Design

This study was intended to exam the students' learning performance of computer programming and their use of self-regulation under two scaffolding supports, one of which, the PS group, was based on the framework of ELCS, in learning computer programming. Thus, two different scaffolding supports (PS group and PR group) were independent variables, and the dependent variables were the students' learning performance of computer programming and their use of self-regulation.

2.3 Procedure

Before the treatment, the students were offered a 40-minute instruction for learning basic operational concepts and skills of Stagecast Creator. And a five-week instruction (respectively designed as two different scaffolding supports: problem-solving support and procedural support) was conducted and viewed as a scaffolding support for the students to learn computer programming in Stagecast Creator. After the instruction, *Computer Programming Test* was adopted to examine the computer programming learning performance of the students in two different groups. Besides, both groups had to self-report their individual use of self-regulation after the five-week instruction through *Self-regulation Learning Questionnaire.*

In *the Problem-solving Support Group*, the students would be guided by the instruction designed according to problem-solving support based on ELCS. Thus, the four steps of the instruction were conducted: observe the problems in the learning task, plan the solving strategy, practice the solving strategy and exam the result and reflect the solving strategy. Figure 2 was the screenshot of the instruction of the problem-solving support.

In *the Procedural Support Group*, the instruction was demonstrated to the students. And the three steps of the instruction were included: explain the concepts of programming, practice programming, and exam the results. Figure 3 was the screenshot of the instruction of the procedural support.

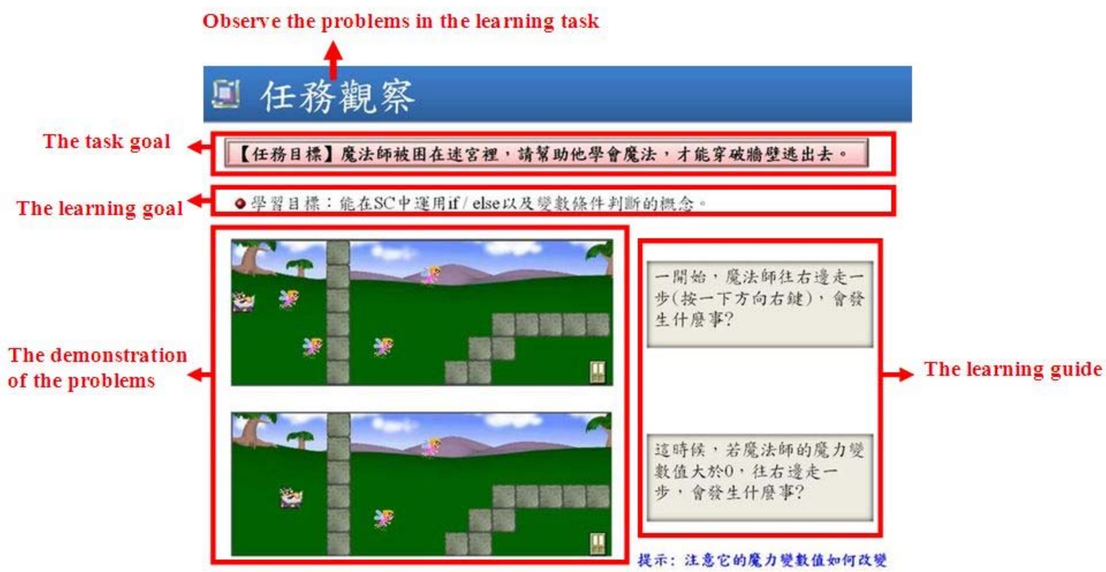

Fig. 2 The instruction of the Problem-solving Support

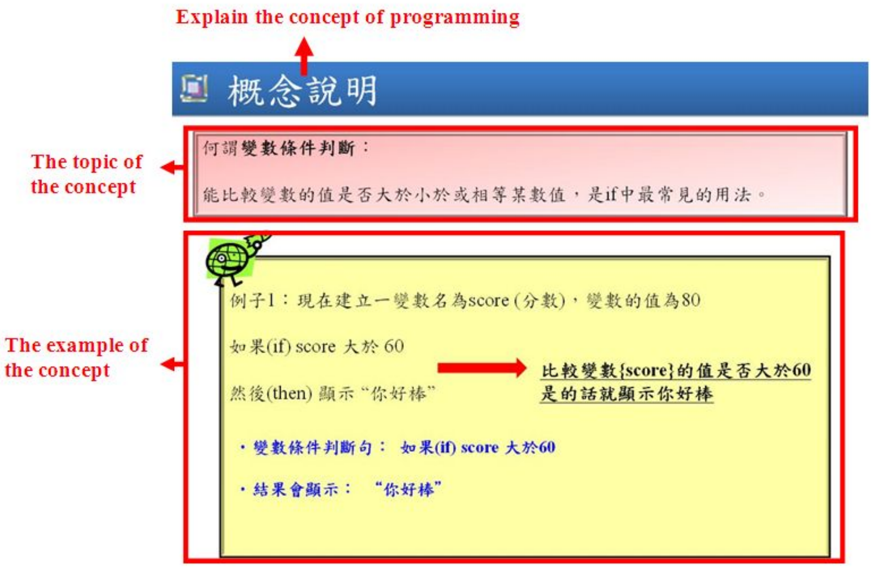

Fig. 3 The instruction of the Procedural Support Group

2.4 Instruments

Computer Programming Test (CPT) was comprised of two parts: basic concepts (9 items) and advance concepts (9 items) for examining the students' learning performance of computer programming. The internal consistency of CPT measured by the Cronbach's α was .88 for all scales. And the content validity of it was also conducted before test.

Self-regulated Learning Questionnaire for Computer Programming (SLQCP), which was modified from *Motivated Strategies for Learning Questionnaire* (MSLQ)[18, 19], was a questionnaire for the students to self-report their use and performance of self-regulation at the end of the treatment for learning computer programming. Three scales of SLQCP were cognitive, motivational, and resource management scale. The 54 items on SLQCP were scored on a 6-point Likert scale, from 1 (not at all true of me) to 6 (very true of me). The cognitive scale consisted of 29 items that assessed students' cognitive strategies (rehearsal, elaboration, organizational, and critical strategies; totally 19 items) and metacognitive strategies (planning, monitoring, and regulating strategies; totally 10 items). And the motivational scale included 16 items: 8 items about students' value components (intrinsic and extrinsic goal orientations) and 8 items about their expectancy components (self-efficacy for learning and expectancy for success). The third scale was resource management contained 9 items, which assessed students' effort management, peer learning, and help-seeking behavior. And Cronbach's α scores of SLQCP was .95.

3. Results

The result shows two parts: one is the students' learning performance of computer programming, and the other is their use of self-regulation.

The mean scores of the learning performance in CPT between PS group and PR group are shown in Table 1. In both of the basic and advance concepts in CPT, the PS group scores higher than the PR group.

Table 1. The mean scores of the learning performance in CPT

Learning performance	Scaffolding support	Mean	SD	*n*
Basic concepts	PS group	8.18	1.468	38
	PR group	7.64	1.871	39
	Total	7.91	1.695	77
Advance concepts	PS group	7.42	2.262	38
	PR group	5.95	2.625	39
	Total	6.68	2.547	77
Total	PS group	15.61	3.515	38
	PR group	13.59	4.166	39
	Total	14.58	3.965	77

As shown in Table 2, one-way ANOVA summary of the learning performance in CPT is used to examine the effect of two different scaffolding supports on the learning performance of computer programming concepts learning. In basic concepts learning, the PS group and the PR group have no significant difference ($F_{(1, 75)} = 2.003$, $p = .161$). However, in advance learning, the PS group is significantly higher than the PR group ($F_{(1, 75)} = 6.936$, $p < .05$). Besides, in total learning performance in CPT, the PS group is also significantly higher than and the PR group ($F_{(1, 75)} = 5.252$, $p < .05$).

Table 2. ANOVA summary of the learning performance in CPT

		Sum of Squares	df	MeanSquae	*F*	Sig.
Basic concepts	Between Groups	5.679	1	5.679	2.003	.161
	Within Groups	212.685	75	2.836		
	Total	218.364	76			
Advance concepts	Between Groups	41.723	1	41.723	6.936*	.010
	Within Groups	451.161	75	6.015		
	Total	492.883	76			
Total	Between Groups	78.186	1	78.186	5.252*	.025
	Within Groups	1116.515	75	14.887		
	Total	1194.701	76			

Note: **p* < 0.05.

The ANOVA summary of the use of self-regulation after the five-week instruction is shown in Table 3. There is only significant difference in help-seeking behavior ($F_{(1, 75)} = 6.813$, $p < .05$), and the PR group (mean = 13.871) is significantly higher than the PS group (mean = 12.000). Besides, in both critical strategies ($p = .061$; PS mean = 18.684; PR mean =20.692) and intrinsic goal orientation ($p = .066$; PS mean = 17.105; PR mean = 18.743), the PR group is nearly significantly higher than the PS group.

Table 3. ANOVA summary of the use of self-regulation after the five-week instruction

		SS	df	Mean Square	F	Sig.
Cognitive Scale	Between Groups	1008.741	1	1008.741	1.972	.164
A.Cognitive Strategies	Between Groups	463.286	1	463.286	2.022	.159
A-1.Rehearsal strategies	Between Groups	20.972	1	20.972	1.372	.245
A-2. Elaboration strategies	Between Groups	44.759	1	44.759	1.657	.202
A-3. Organizational strategies	Between Groups	2.087	1	2.087	.104	.747
A-4.Critical strategies	Between Groups	77.612	1	77.612	3.610	.061
B. Metacognitive strategies	Between Groups	104.788	1	104.788	1.309	.256
B-1 Planning strategies	Between Groups	14.947	1	14.947	.978	.326
B-2. Monitoring strategies	Between Groups	.313	1	.313	.058	.810
B-3. Regulating strategies	Between Groups	33.768	1	33.768	2.021	.159
Motivational Scale	Between Groups	.148	1	.148	.001	.980
C. Value components	Between Groups	29.701	1	29.701	.520	.473
C-1. Intrinsic goal orientation	Between Groups	51.660	1	51.660	3.469	.066
C-2. Extrinsic goal orientation	Between Groups	3.020	1	3.020	.143	.706

D. Expectancy	Between Groups	25.654	1	25.654	.341	.561
D-1. Self-efficacy for learning	Between Groups	10.392	1	10.392	.286	.594
D-2.Expectancy for success	Between Groups	3.390	1	3.390	.301	.585
Resource management scale	Between Groups	76.883	1	76.883	1.543	.218
E. Effort management	Between Groups	1.213	1	1.213	.223	.638
F.Peer learning	Between Groups	.297	1	.297	.017	.897
G.Help-seeking behavior	Between Groups	67.433	1	67.433	6.813*	.011

Note: $^{*}p < 0.05$.

4. Conclusion and Discussion

The aim of the study is mainly to explore the students' use of self-regulation and their learning performance of computer programming under the framework of ELCS in learning computer programming. This framework emphasizes the experiential learning processes and attempts to facilitate elementary school learners' acquisition of computer programming learning through scaffolding support based on ELCS.

And computer programming learning is abstract and complex concept for school students to understand, apply, and even transfer it in different learning contexts. According to the attributes of this instructional framework, the abstract and complex concepts and knowledge will be visualized, concreted and simplified through interactive animations, simulations and instructional games.

According to the results of the students' learning performance of computer programming through CPT, the PS group has better performance than the PR group. This proves that problem-solving scaffolding support enhances the elementary school students' learning in both the advance and entire concept learning of computer programming. Many researchers also find out that learners are merely aware of the problems that can be solved by a computer and the benefits to be had from using programming, so it is important to provide an appropriate support to guide and lead them to express their problem-solving strategies in order to progress smoothly to the formation of the appropriate code [1, 9]; Brooks, 1999; [7, 8].

On the other hand, in the students' use of self-regulation, the result shows that the PR group tries to ask others for help frequently higher than the PS group. Computer programming is always treated as a complex task, and learners need to have access to understanding of the task, method finding, coding, testing and debugging of the resulting program (Brooks, 1999). But for novices, without using a suitable support (for example, providing daily-life problems to solve), it would be hard to perceive the target attributes of the abstract concepts and reflect their observations to link with related prior knowledge [2]. The students in the PR group need to ask more help to assist themselves to understand and comprehend computer programming concepts.

In the future, learners' learning performance in computer programming project will be the target to investigate further. And in the process of the project, their use of SRL will also need to explore deeper to find out the fluctuation of cognition, motivation and attitude under the instructional framework of ELCS in learning computer programming.

References

[1] Brooks, R. (1999). Towards a theory of the cognitive processes in computer programming. *International Journal of Human Computer Studies, 51*, 197–211.

[2] Chen, M. P., Wu, C. C., Chiu, C. H., Lee, H. L. & Chen, M. C. (2009). Teaching abstract computer science concepts to high school students: A multiple-perspective approach. *International Journal on Digital Learning Technology, 1*(4), 266-286.

[3] Crews, T., & Butterfield, J. (2002). Using technology to bring abstract concepts into focus: A programming case study. *Journal of Computing in Higher Education, 13*(2), 25-50.

[4] Dresel, M., & Haugwitz, M. (2008). A computer-based approach to fostering motivation and self-regulated learning. *Journal of Experimental Education, 77*(1), 3-18.

[5] Flavell, J. H. (1979). Metacognition and cognitive monitoring. *American Psychologist, 34*(10), 906-911.

[6] Kolb, D. A. (1984). *Experimental learning*. Englewood Cliffs NJ: Prentice-Hall.

[7] Komis, V. (2001). A study of basic programming concepts within a constructivist teaching approach. *Themes in Education, 2*(2-3), 243-270.

[8] Kordaki, M. (2007). Modeling and multiple representation systems in the design of a computer environment for the learning of programming and C by beginners. In T. Bastiaens, S. Carliner (Eds.), *Proceedings of world conference on e-learning in corporate, government, healthcare & higher education (e-learn 2007), October, 15–19, Quebec, Canada, USA* (1634–1641). Chesapeake, VA: AACE.

[9] Lemone, K. A. & Ching, W. (1996). Easing into C: Experiences with RoBOTL. *ACM SIGCSE Bulletin, 28*(4), 45-49.

[10] Malik, M. (2000). Technical opinion: On the perils of programming. *Communications of the ACM, 43*(12), 95-97.

[11] Mayer, R. E. (1992). *Thinking, problem solving, cognition*. New York: W. H. Freeman.

[12] McGill, T. J & Volet, S. E. (1997). A conceptual framework for analyzing students' knowledge of programming. *Journal of Research on Computing in Education, 29*(3), 276-297.

[13] Milbrandt, G. (1999) Computer problem-solving and programming: some number theory examples. *Journal of Computer Science Education, 13*(3&4), 18-26.

[14] Moos, D., & Azevedo, R. (2008). Exploring the fluctuation of motivation and use of self-regulatory processes during learning with hypermedia. *Instructional Science, 36*(3), 203-231.

[15] Palumbo, D. B. (1990). Programming Language/Problem-Solving Research: A Review of Relevant Issues. *Review of Educational Research, 60*(1), 65-89.

[16] Pintrich, P. R. (2000). The role of goal orientation in self-regulated learning. In Boekaerts, M., Pintrich, P. R., and Zeidner, M. (eds.), *Handbook of Self-Regulation*, Academic Press, San Diego, CA, 451-502.

[17] Pintrich, P. R., Smith, D. A. F., & McKeachie, W. J. (1989). *A manual for the use of the Motivated Strategies for Learning Questionnaire (MSLQ)*. MI: National Center for Research to Improve Postsecondary Teaching and Learning.

[18] Pintrich, P. R., Smith, D. A. F., Garcia, T., & McKeachie, W. J. (1991). *A manual for the use of the Motivated Strategies for Learning Questionnaire (MSLQ)*. MI: National Center for Research to Improve Postsecondary Teaching and Learning.

[19] Pintrich, P. R., Smith, D. A., Garcia, T., & McKeachie, W. J. (1993). Reliability and predictive validity of the Motivated Strategies for Learning Questionnaire (MSLQ). *Educational and Psychological Measurement, 53*(0), 801-813.

[20] Soloway, E. (1993). Should we teach students to program? *Communications of the ACM, 36*(10), 21-24.

[21] Thomas, R. A. & Upah, S. C. Jr. (1996) Give programming instruction a chance. *Journal of Research on Computing in Education, 29*(1), 96-107.

[22] Tucker, A., Deek, F., Jones, J., McCowan, D., Stephenson, C., & Verno, A. (2006). *A model curriculum for K-12 computer science: Final report of the ACM Task Force Curriculum Committee* (2nd ed.). NY: Association for Computing Machinery.

[23] West, M., & Ross, S. (2002). Retaining females in computer science: A new look at a persistent problem. *JCSC*, *17*(5), 1-7.

[24] Yuen, A. H. K. (2006). Learning to program through interactive simulation. *Educational Media International, 43*(3), 251-268.

[25] Zimmerman, B. J. & Martinez-Pons. M. (1986). Development of a structured interview for assessing student use of self-regulated learning strategies. *American Educational Research Journal, 23*(4), 614-628.

[26] Zimmerman, B. J. (1986). Becoming a self-regulated learner: Which are the key subprocesses? *Contemporary Educational Psychology, 11*(4), 307-313.

[27] Zimmerman, B. J. (2002). Becoming a self-regulated learner: An overview. *Theory into Practice,41*(2), 64-70.

[28] Zimmerman, B. J., & Martinez-Pons, M. (1988). Construct validation of a strategy model of student self-regulated learning. *Journal of Educational Psychology, 80*(3), 284-290.