Effects of Self-Efficacy and Feedback Strategies on Debugging Activities

100學年度上學期
教學行動研究

資訊管理系 藍友烽

Introduction

- Due to the rapid development of information technology, computer programming will become more widespread as more schools add computer classes to their curriculums.

- Computer programming skills constitute one of the core competencies of a graduate from many disciplines, such as engineering and computer science, are expected to possess (Law, Lee, & Yu, 2010).

Introduction

- Learning computer programming is difficult and students often have some difficulty confirming original constructs such as counters, variables, loops, and conditions in their initial solution (Eckerdal, 2009; Lahtinen, Ala-Mutka, & Järvinen, 2005).

- Computer programming courses are perceived as uniquely demanding, characterized by the large amount of exercises students are expected to intensively practice to develop good programming skills and gain experience in debugging (Lam, Chan, Lee, & Yu, 2008).

Introduction

- Debugging has been known to account for more than 50% of the time and effort spent in the development of a computer program (Myers, 1997; Ward, 1988).

- Although much research has been devoted to the design and implementation of software systems that aid learning of computer programming, there seems to be few studies focused on debugging practice activities.

Introduction

- Shneiderman (1980) and Pope (1989) pointed out there are some relationships between personal characteristics and how to do well in computer work.

- Sariya (1991) also indicated the personal attitude toward computers is an important factor affecting success in computer education.

Introduction

- When developing debugging skills, individual differences between potential students should be considered.
- Particularly, self-efficacy has a stronger effect on academic performance than other motivational beliefs (Pintrich & DeGroot, 1990; Pintrich & Schunk, 2002).
- The purpose of this study was to develop a debugging practice environment and provide various feedback strategies to improve students' debugging performance.

Literature review

- Feedback strategies
- Debugging learning activities

Feedback strategies

- Feedback is crucial for promoting efficient learning, but is often a neglected factor with computer-based training programs (Clariana, Ross, & Morrison, 1991).

- Feedback has been argued to play an important role in learning, and it influences performance in different ways depending on how it is provided (Hattie & Timperley, 2007).

Feedback strategies

- Melis (2005) pointed out giving feedback influences the learners' ability to progress in problem solving and learning, affecting their motivational and affective state.

- Research suggested feedback can help individual students correct misconceptions, reconstruct knowledge, support metacognitive processes, improve academic achievement, and enhance motivation (Clark & Dwyer, 1998; Foote, 1999; Warden, 2000; Zimmerman & Martinez-Pons, 1992).

Feedback strategies

- The interactive capabilities of modern information technology can increase the range of feedback strategies and be implemented in computer-based training (Hannafin, Hannafin, & Dalton, 1993; Narciss & Huth, 2006).

- A previous study (Halabi, 2006) found rich feedback was significantly more useful for students with no prior knowledge.

Feedback strategies

- Most importantly, Melis (2005) indicated the actual value of feedback depends on how well its type, content, and form match the characteristics of the instructional context and the learner.

- Therefore, providing learners with suitable types of feedback and feedback presentations should be considered in computer-based training to enhance learning performance.

Debugging learning activities

- Debugging is a continual process of hypothesis generation and verification (Araki, Furukawa, & Cheng, 1991).

- The final goal of the debugging process is to remove defects from computer programs (Chmiel & Loui, 2004).

- Specifically, it is a process of locating the exact position of the error and correcting it after the existence of the error is verified by testing (Vessey, 1986).

Debugging learning activities

- Computer programming classes often concentrate on teaching programming language syntax, problem analysis, and writing programs to solve problems.
- Class time is seldom allocated to debugging practice activities.
- Debugging training is even more important for novice programmers. (Lee & Wu, 1999).

Methodology

- Implementation of Debugging Feedback System
- Research model
- Participants
- Procedures
- Measurements

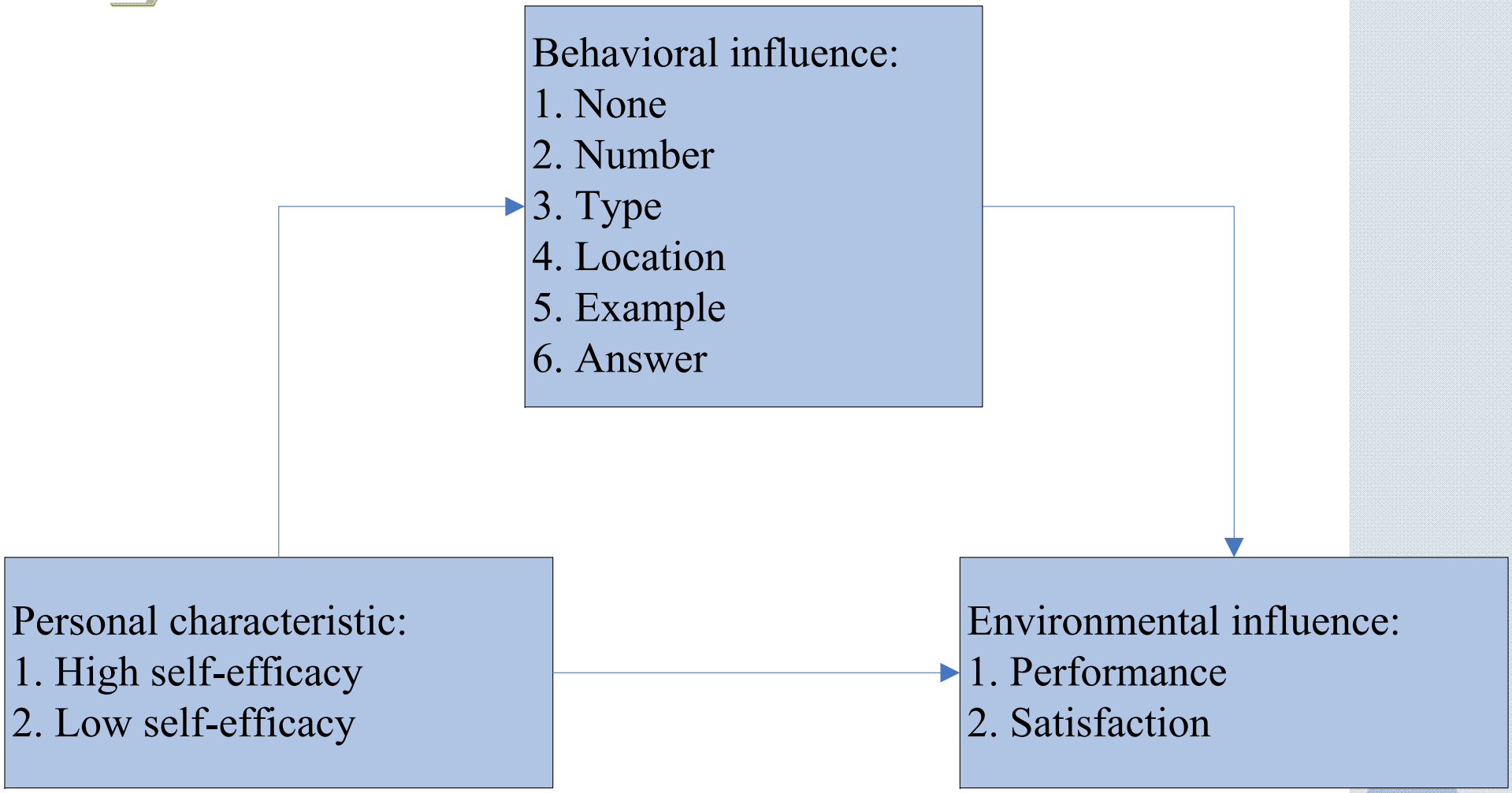
Implementation of Debugging Feedback System

- In order to help students improve debugging skills and give them an environment for practice, this study developed Debugging Feedback System for students.
- The proposed system mainly focuses on debugging exercises for loop constructs.
- Winslow (1996) indicated sophisticated material is taught to CS1 students when study after study has shown they do not understand basic loops.

Implementation of Debugging Feedback System

- The most often committed novice-programming errors associated with loops are collected from the instructor's teaching experience.
- The corresponding programs are written in C and all programs are embedded with one or more of the common errors.
- Available at http://140.130.33.84/debug

Research model

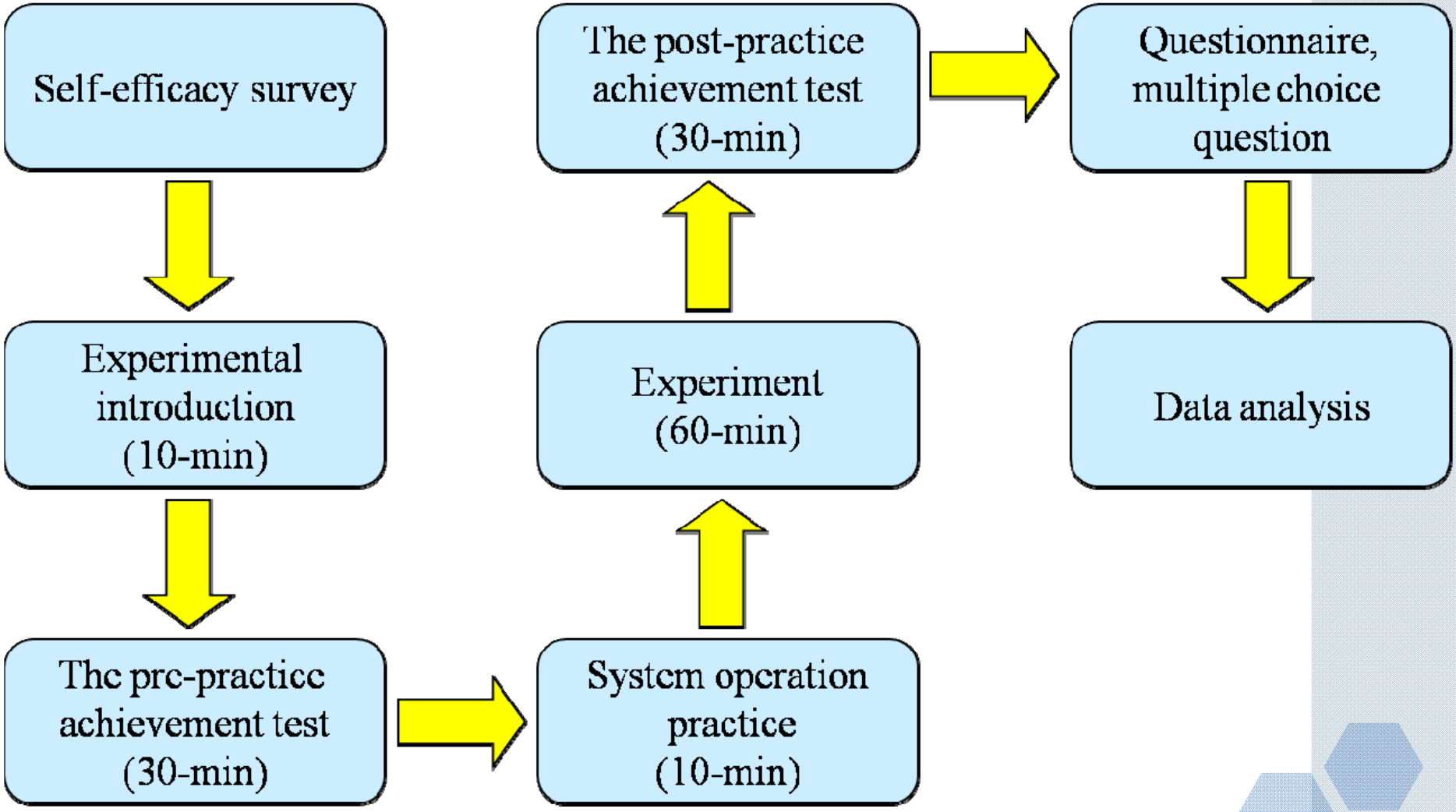


Participants

- The participants for this research were sixty-two first-year college students majoring in the Information and Management Department.

- The title of the experiment course was “Introductory Computer Programming”.

Procedures


Self-efficacy survey
Experimental introduction (10-min)
The pre-practice achievement test (30-min)
System operation practice (10-min)
Experiment (60-min)
The post-practice achievement test (30-min)
Questionnaire, multiple choice question
Data analysis

Data analysis and results

The paired samples t-test of pre-test and post-test scores on debugging achievement

	N	Mean	SD	t	Sig.
Pre-test	58	48.345	15.829	-13.809	.000*
Post-test	58	67.828	12.920		

*$p<.05$

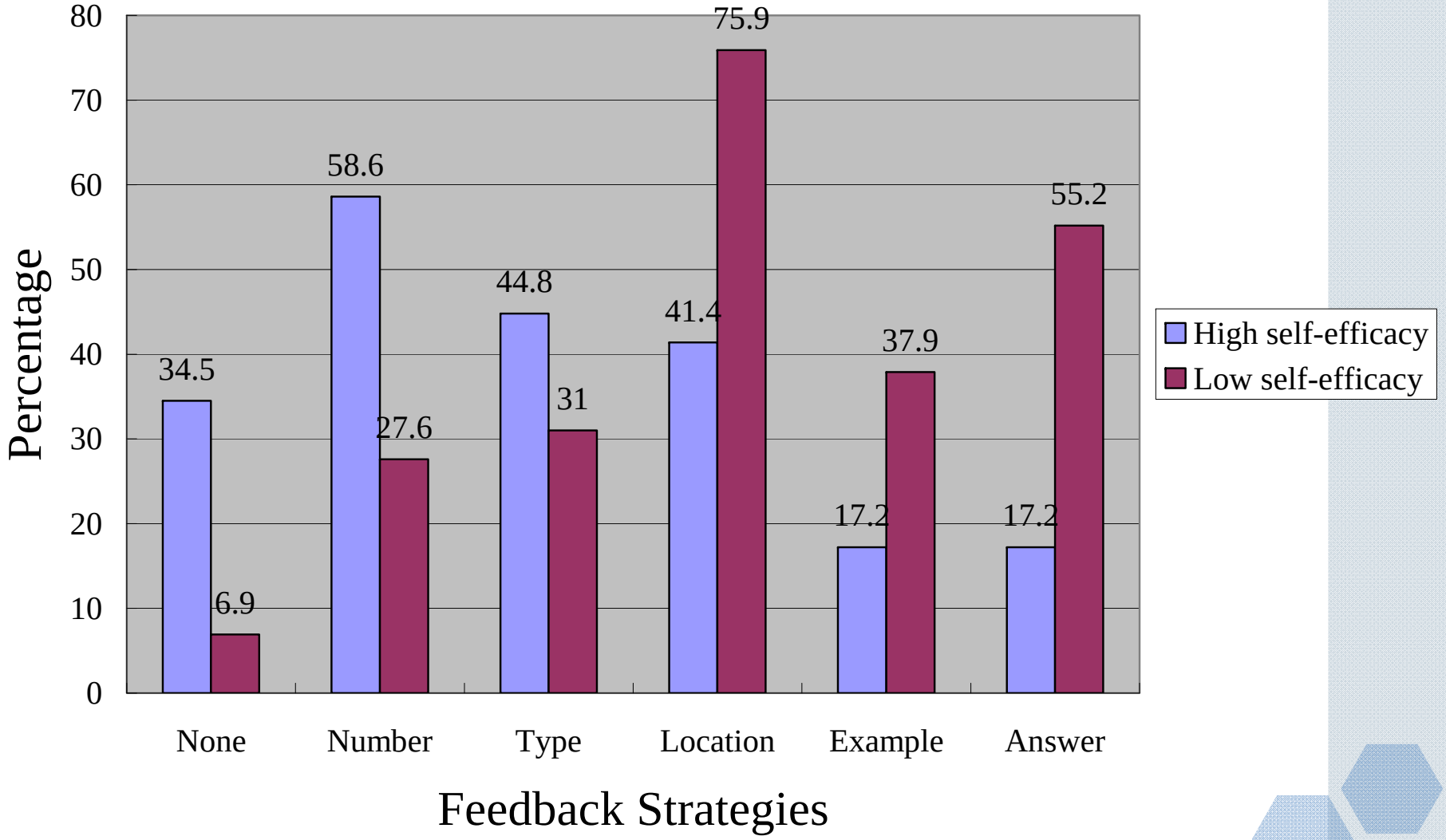
Data analysis and results

The paired samples t-test of pre-test and post-test scores for high self-efficacy and low self-efficacy students

	High self-efficacy					Low self-efficacy				
	N	Mean	SD	t	Sig.	Mean	SD	t	Sig.
Pre-test	29	49.862	14.431	-8.465	.000*	46.828	17.234	-11.374	.000*
Post-test	29	68.621	12.120			67.035	13.842		

*$p<.05$

Data analysis and results



Self-efficacy differences in feedback strategies selection when not considering weighted score.

Data analysis and results

Perceived usefulness of the proposed system

#	Question	SD	D	U	A	SA	Avg.
1	I thought that using the proposed system is efficient for me to quickly detect bugs.	0 0.00%	1 1.72%	4 6.90%	33 56.90%	20 34.48%	4.24
2	I thought that using the proposed system can improve my performance in the program debugging activities.	0 0.00%	2 3.45%	7 12.07%	20 34.48%	29 50.00%	4.31
3	I thought that using the proposed system is useful in the program debugging activities.	0 0.00%	2 3.45%	5 8.62%	22 37.93%	29 50.00%	4.34
4	On the whole, I thought that using the proposed system can find more bugs for me in the program debugging activities.	0 0.00%	5 8.62%	8 13.79%	21 36.21%	24 41.38%	4.10

Note: SD: Strongly Disagree, D: Disagree, U: Undecided, A: Agree, SA: Strongly Agree.

Data analysis and results

Perceived ease of use of the proposed system

#	Question	SD	D	U	A	SA	Avg.
5	It would be easy for me to become skillful at using the proposed system.	0 0.00%	5 8.62%	8 13.79%	28 48.28%	17 29.31%	3.98
6	I thought that learning to operate the proposed system is easy for me.	0 0.00%	0 0.00%	5 8.62%	25 43.10%	28 48.28%	4.40
7	I thought that using the proposed system is easy for me to select what feedback strategies I want.	1 1.72%	3 5.17%	10 17.24%	26 44.83%	18 31.03%	3.98
8	On the whole, I thought that the proposed system easy to use.	0 0.00%	2 3.45%	11 18.97%	19 32.76%	26 44.83%	4.19

Note: SD: Strongly Disagree, D: Disagree, U: Undecided, A: Agree, SA: Strongly Agree.

Discussion and Conclusion

- The results of this study indicated debugging performance was significantly improved by our proposed system.

- Even though both high self-efficacy or low self-efficacy students chose the "None" feedback strategy among six feedback strategies when considering weighted score, this strategy was not necessarily successful for low self-efficacy students.

Discussion and Conclusion

- This finding might be due to the fact the "None" feedback strategy among the six feedback strategies has the highest weighted score, but the students were still new to the “Computer Programming” course and thus might not have had enough performance experience or information to judge their efficacy in the domain.

Discussion and Conclusion

- A finding also needs to be discussed is the total successful debugging rate was higher than the total failed debugging rate for both high and low self-efficacy students during debugging practice activity.

- This may indicate the problems were generally not too difficult for the students, and more elaborate feedback strategies could have useful in the most complex problems.

Discussion and Conclusion

- Regarding selection of feedback strategy not considering weighted score, high self-efficacy students used more low-level feedback strategies, such as "None" and "Number" of feedback strategies, compared to low self-efficacy students.

- Low self-efficacy students used more high-level feedback strategies, such as "Location" and "Answer" of feedback strategies, compared to high self-efficacy students.

Discussion and Conclusion

- High self-efficacy students used more low-level feedback strategies than high-level feedback strategies and low self-efficacy students used more high-level feedback strategies than low-level feedback strategies.

- Based on the evidence, the degree to which a specific feedback strategy is preferred to solve problems seems to depend on the level of weighted score of the feedback strategies.

Discussion and Conclusion

- According to the results of the questionnaire, students had good technology acceptance for the proposed system and they felt satisfied with it.

- The practical implications of the results are different self-efficacy students have different feedback strategy preferences and the developer and designer of an adaptive debugging practice environment could adopt suitable feedback strategies to support the students having different self-efficacy in debugging activities.

Discussion and Conclusion

- In conclusion, an adaptive learning system should be applied in programming courses to help students develop their debugging skills.

- We believe this kind of instruction can lead to good learning achievements, and students will not feel helpless and frustrated during the program debugging learning process.

Thank You!