The Social Sciences 6 (6): 495-501, 2011
ISSN: 1818-5800

The Effects of the Instructional Process on the Logical Thinking Skills of Software Developers

Nantaporn Booneka and Paiboon Kiattikomol
Learning Innovation in Technology Program, Faculty of Industrial Education and Technology,
King Mongkut's University of Technology, Thonburi, Thailand

Abstract: The objectives of this study relate to logical thinking skills for software developers. It identifies the content and instructional process in developing logical thinking skills and evaluates the content and instructional process. Interviews with twenty participants revealed that the content of logical thinking skills includes general problems, probability problems, arithmetic expressions and arithmetic and logical comparisons. The instructional process of developing logical thinking skills for software developers is comprised of four steps. These include understanding, planning, executing and evaluation. The researchers then asked eighteen experts in logical thinking skills, learning and instruction to evaluate the content of the logical thinking and instructional process. The results showed that the overall evaluation was good (4.28/5), the content evaluation was good (4.21/5) and the instructional process evaluation was good (4.36/5). The results of this study may be of interest to institutes and to instructors who wish to rely on the content and instructional process to develop curriculum focused on logical thinking skills.

Key words: Logical thinking skills, learning outcome, instructional process, software developer, arithmetic expressions, probability problems

INTRODUCTION

The skill of logical thinking is an important competency for software developers. Studies by Booneka and Kiattikomol (2008) show that logical thinking skills is one of the twelve necessary competencies for software developers in Thailand. They ranked logical thinking as number three out of 12 competencies. Similarly, studies by Haines showed that logical thinking skills play an important role in achievement in biology and in computer programming and mathematics for computing. Logical thinking skills require abilities such as understanding the structure of problems adequately, thinking reasonably about problems and expressing logically (Kawamoto and Arai, 2008).

With logical thinking, the learner solves the problem by doing various mental practices or reaches principles or rules by engaging in abstraction and generalization (Yaman, 2005). Greeno (1989) found that successful thinking in specific subject-matter domains uses knowledge that is specific to the domains. In addition, Beyer (2008) argues that thinking skills are not content free or ill-defined phenomena. This implies that logical thinking skills is a specific area of competence.

Tobin and Capie (1981) developed the Test of Logical Thinking skills (TOLT) which has five measures, namely; proportional reasoning, the control of variables, probabilistic reasoning, correlational reasoning and combinatorial reasoning. Likewise, The Group Assessment of Logical Thinking (GALT), developed by Roadrangka *et al.* (1982), measures six logical operations. These are conservation, proportional reasoning, controlling variables, combinatorial reasoning, probabilistic reasoning and correlational reasoning. These two tests have the same five measures. Implied logical thinking means to perform at least five measures, namely controlling variables, proportional reasoning, probabilistic reasoning, combinatorial reasoning and correlational reasoning. These two tests were used to measure logical thinking skills in any area whereas logical thinking skills that apply specifically for software developers has not been identified in any research.

Mayer (2008) argues that learning in the content areas such as learning to read fluently, learning to write, learning science or learning mathematics is different. Each area has a different cognitive process. Furthermore, logical thinking skills are similar to learning mathematics because as Ralston (2005) argues, the logical thinking inherent in all mathematical thought is so akin to the logical thinking required in all software development. Additionally both TOLT and GALT also rely on mathematical thinking. Marcou and Lerman (2006), citing

Corresponding Author: Nantaporn Booneka,Learning Innovation in Technology Program,
Faculty of Industrial Education and Technology,King Mongkut's University of Technology, Thonburi, Thailand

some of models of Mathematical Problem Solving (MPS), refer to the four steps of Polya and the five step cognitive self regulatory strategy of Verschaffel *et al*. (1999). The four steps of Polya has been adapted by Barnes *et al*. (1997) to a programming context and also investigated how it fits the imperative and functional programming paradigms. These steps comprise understanding, design, writing and review. Similarly, the study by Chang *et al*. (2006) also designed a computer-assisted system named MathCAL based on the four steps of Polya and showed that the system improves the performance of students with low problem-solving ability.

The five steps cognitive self regulatory strategy of Verschaffel *et al*. (1999) implemented a learning environment for teaching and learning how to model and solve mathematical application problems for upper elementary school. The five steps comprise:

- Building a mental representation of the problem by drawing a picture, making a list, making a scheme or a table, distinguishing relevant from irrelevant data, using your real-world knowledge
- Deciding how to solve the problem by making a flowchart, guess and check, look for a pattern and simplify the number
- Execution of the necessary calculations
- Interpretation of the outcome and formulation of an answer
- Evaluation of the solution

The four-mental processes for solving mathematics problems of Mayer (2008) are similar. Learners solve mathematical problems by means of problem translation, problem integration, solution planning and monitoring and solution execution. The content and the strategy for solving problems by logical thinking skills has not been identified in any research. The study reported on in this study had three objectives:

- Identification of the content of logical thinking for software developers
- Identification of an instructional process related to logical thinking skills
- Evaluation of the content of logical thinking and the instructional process for developing logical thinking skills

MATERIALS AND METHODS

Identifying the content of logical thinking skills

Participants: The research team interviewed 20 people working as software developers, system analysts, Information Communications Technology (ICT) lecturers and ICT managers from fifteen organizations, nine government organizations and six private organizations in Bangkok, Thailand. The 1st step involved contacting by mail the heads of each organizations to inform them of the study and its purpose and to formally invite their participation. The organizations were also requested to select the superior or most effective performers in the area of developing software. The organizations then identified individuals who were qualified and who the researchers could interview. The outline of interview was forwarded to the participants and appointments were set up.

Instrument: The face-to-face interview form Slavin (2007) was made up of open ended questions as follows:

- What is the content of logical thinking skills?
- What is the content of logical thinking skills for software developers?
- Why are the contents that you propose so important?
- Can you show examples of the problems that software developers without logical thinking skills face?
- How do you solve the problems that software developers face?

Each interview lasted from one to one and half hours and totaled 27 h. The interviews were audio recorded.

Data analysis: Six steps were used in analyzing the qualitative data (Creswell, 2008):

- Preparing and organizing the data
- Exploring and coding the database
- Describing the findings and formulating themes
- Representing and reporting the findings
- Interpreting the meaning of the findings
- Validating the accuracy of the findings

Identify the instructional process of logical thinking skills: The purpose of this part was to identify the instructional process of logical thinking skills for software developers. Three models of mathematical problem solving were adapted by the researchers. These were:

- The four processes of Polya
- The five processes of Verschaffel *et al*. (1999)
- The four processes of Mayer (2008)

The researchers then synthesized the instructional process of logical thinking skills for software developers based upon aspects that were the best and the most suitable for this study.

Table 1: Rating scale for evaluation

Numerical rating	Descriptive rating
4.50-5.00	Very good
3.50-4.49	Good
2.50-3.49	Fair
1.50-2.49	Poor
1.00-1.49	Very poor

Evaluate the content and the instructional process for developing logical thinking skills: The purpose of the study was to evaluate the content of logical thinking skills and the instructional process of logical thinking skills.

Participants: Eighteen experts who had >10 years experience in logical thinking skills, learning and instruction were asked to evaluate the logical thinking content and instructional process. Each expert was contacted by mail with the letter of cooperation of the coordinating institution of the researchers.

Instruments: The experts evaluated the content and instructional processes by completing an evaluation process. Firstly, focus on demographic information such as gender, age, position, years of experience. The researchers collected demographic information about the respondents in order to ensure that experts had >10 years experience in logical thinking skills, learning and instruction.

Then, the experts were asked to evaluate two things, namely; the content and instructional processes. The experts ranked each one on a scale of 1-5. The content part comprised four outcomes, namely; a general problem, a probability problem, an arithmetic expression and an arithmetic and logical comparison. For each outcome, the researchers provided the objectives, a brief of the content and the learning outcome after finishing each module. The instructional process part comprised four steps, namely; understanding, planning, executing and evaluation. For each instructional process, researchers provided the concepts, guidelines, procedures and sample activities that the instructors used to teach the learners.

Data analysis: Once the experts had completed their evaluation using their 5-point scale, the researchers then calculated the average of each learning outcome and the average of each instructional process. These were then translated into descriptive measures as shown in Table 1. The overall average of both the learning outcomes and instructional process were also calculated.

RESULTS

The content of logical thinking skills

Outcome 1: The interviewees focused on general mathematical problems such as ratio, percentage, interest. These are the kind of problems everyone faces in daily life. The interviewees also argued that if the learners can solve these problems, it is easy for them to design algorithms. In relation to this, the interviewees commented as follows:

When I taught the programming course, some of students could not solve simple problem such as percentage, interest, ratio or rate of work. When I asked my students to determine if a number is an odd or even number, they could determine it but they could not write an algorithm to solve this problem. I think basic mathematics is very important for my students. When I have new programmers or intern student before starting work I ask them to develop a small program such as calculating income and commission. It surprises me that some of them used wrong formulas to calculate, some of them do not know where to start. I think that it is the beginning steps that are very important. After they finish, I also ask them to check the answer with each other and discuss together whether they have similar or different answers. Next, I ask them to develop a small program from daily life such as withdrawing cash from an ATM or shopping at a supermarket.

After completing this outcome, learners will be able to understand and apply knowledge to solve general problems including designing algorithms to solve the problem. Learners can apply knowledge to solve programming problems.

Outcome 2: The interviewees focused on the probability of something happening. The chance that it will happen is expressed as a fraction or percentage. The interviewees argued that learners should know all possible events that will result from particular actions. In relation to this, the interviewees commented as follows:

My students did not identify all possible events that could occur such as when I asked them if there are red, white and yellow balls in the box what is the likelihood that I will pick up a red ball? One thing that I emphasize with my new staff when they design programs is to list all possible ways for each case. For example when people withdraw cash from an ATM, the program must check all of the possible conditions that could occur. For example, people withdraw less than the balance or people withdraw an amount equal to the balance or people withdraw more than balance. The program covers all possible cases and then sets criteria to allow what is permitted. Each bank may have a different of conditions in place.

I think probability is important because when students design an algorithm they must identify the possibility that it will occur. If students or developers do not identify a possible event it will result in the design of the wrong algorithm. After completing this outcome, learners will be able to understand probability, express knowledge of the fact that an event will occur or has occurred, apply probability to solve general problems and apply probability to a programming problem.

Outcome 3: The interviewees focused on how to convert problems to an arithmetic expression because when learners face problems, they always do this in sentences. Learners must convert these sentences to arithmetic expressions and then solve the problems. When learners solve arithmetic expressions, they follow the order of arithmetic operators.

The interviewees emphasized that two consecutive steps are very important. First, it is necessary to convert problems to an arithmetic expression. Second, the arithmetic expression needs to be solved by following the order of arithmetic operators. In relation to this outcome, the interviewees commented as follows:

My students did not convert sentences to an arithmetic expression. For example, X is a number which when added to 10 equals 45. Some of students could not convert this to an arithmetic expression.

When I asked my student to find the results of the arithmetic expression: X = 20 - 2×3^2, half of them got the wrong result. It means that the students did not know the order of the arithmetic process. Students must know how to translate from a sentence to an arithmetic expression because most problems are described in text. They must know the order of precedent or the order of arithmetic processing. For example, construct an arithmetic expression to find out the area of a square if the width equals X and the length equals Y. Not even half of the students could solve this problem.

Some of my new staff cannot convert a business process to computer programming or algorithm. Before letting them design any algorithm we provide them with the business processes, rules or regulations that are related to the program and make sure that they understand. If they do not understand clearly, it will affect the algorithm that they design.

One important thing is that business rules can change at anytime so developers must follow those changes and modified programs to align with business processes.

The interviewees emphasized this outcome and argued that this content is very important and is the basis of any programming course. Some of interviewees suggested that this kind of content should be taught to students before they are taught other programming languages. After completing this outcome, learners will be able to convert the problem to an arithmetic expression, understand the order of processing arithmetic operations, apply the arithmetic operation to solve related problems and construct an arithmetic expression.

Outcome 4: The interviewees focused on the how to set the criteria of an algorithm when it needs to make a decision or involves looping. The basic criteria are comprised of only one arithmetic comparison such as greater than, less than, equal, not equal, greater than or equal, less than or equal.

The complex criteria are comprised of >1 arithmetic comparison. Each arithmetic comparison is combined together with a logical comparison such as AND, OR, NOT. In relation to this outcome, the interviewees commented as follows:

Some of my student confused arithmetic and logical comparison, especially when both arithmetic and logical comparisons were combined together.

The arithmetic and logical comparisons are important parts when students design algorithms for selection or decision and looping.

One problem faced by my junior programmers is that when they set compound or complex criteria in programs they are sometimes confused and when the programs run, they produce an error.

Most of the interviewees emphasized that if students or programmers clearly understand both arithmetic and logical comparisons it will benefit them when they design algorithms. After completing this outcome, learners will be able to understand arithmetic comparisons, understand logical comparisons and apply knowledge of arithmetic and logical comparisons to solve related problems.

The instructional process of logical thinking skills: The instructional process of logical thinking skills have been adapted and synthesized from three models of mathematical problem solving as described before. The learning process for the development of logical thinking skills comprises four steps as follows:

Step 1 (Understanding): In this step, the instructor will teach learners to understand the problem by translating and/or restating the sentence, highlighting the important point in the problem, identifying the answer required, eliminating sentences that are not related to solving the problem and encouraging the learners to solve the problem. In this step, the instructor may draw a picture, make a list or make a scheme or a table to assist the learners to understand.

Step 2 (Planning): In this step, the instructor will teach learners to recognize the problem types, look around to

Table 2: Results of the evaluation of learning outcomes

Outcomes	Mean	Level of evaluation
General problems	3.72	Good
Probability problems	3.94	Good
Arithmetic expressions	4.50	Very good
Arithmetic and logical comparisons	4.67	Very good
Average of Mean	4.21	Good

Table 3: Result of the evaluation of instructional process

Instructional process	Mean	Level of evaluation
Understanding	4.50	Very good
Planning	4.61	Very good
Executing	4.28	Good
Evaluation	4.06	Good
Average of Mean	4.36	Good

find related problem that learners have already solved, distinguish the relevant and irrelevant information for the solution, break the problem into sub-goals and monitor what the learners are doing. In this step, the learners will make a flowchart, guess and check and seek a pattern to solve the problem.

Step 3 (Executing): In this step, the instructor will teach the learners to examine the solution obtained, ensure that learners check and see clearly the purpose of each step of the design and then apply the rule of mathematics to execute the necessary calculations or parts of the result separately, put the parts of result together, interpret the outcome and formulate or find out the answer.

Step 4 (Evaluation): In this step, the instructor will teach learners to reflect and look back on the finished result and then provide a means to build up the store of experience that is to be applied in future planning steps or derive the solution differently and then examine ways to improve the solution.

Evaluation of the content and instructional process of logical thinking skills: The overall evaluation of the content and instructional process revealed an overall result of good (4.28). The content evaluation revealed good (4.21). When each item that was part of the evaluation was considered, it was found that each was at the good and very good level as shown in Table 2.

The result of the instructional process evaluation was at the very good level (4.36). When the researchers considered each evaluation item they found that each evaluation item was at the good and very good level as shown in Table 3.

DISCUSSION

The content of logical thinking skills comprised four outcomes, namely; general problems, probabilistic problems, arithmetic expressions and arithmetic and logical comparisons. Both the TOLT and GALT tests use the same five measures, namely; proportional reasoning, control of variables, probabilistic reasoning, correlational reasoning and combinatorial reasoning.

The similarity between the content of this research and the TOLT and GALT tests is considered in two parts. Outcome 1 (general problem) covers proportional reasoning. This outcome focuses on ratio, percentage and interest but proportional reasoning measures only ratio problems.

Outcome 2 (probability problem) is very close to probabilistic reasoning. Both outcome 2 and TOLT and GALT measure probability. Additionally, proportionality was defined as a major concept because it is the basis for fractions and unit-price analysis in arithmetic, percentages in algebra, slopes of lines and similar figures in geometry (NCTM, 1989; Greenes, 2009). The content of logical thinking skills for software developers is comprised of only four outcomes but both TOLT and GALT include the same five measures.

The two outcomes of the content that are different from both TOLT and GALT are arithmetic expressions and arithmetic and logical comparisons while TOLT and GALT measure control of variables, correlational reasoning and combinatorial reasoning.

In this study, we focused on logical thinking only for software developers and only on the interviewee's perspective. Some of the interviewees argued that people who want to be software developers must understand basic mathematics and be able to solve problem or design algorithms. They also emphasized that arithmetic expressions and arithmetic and logical comparisons are very important concepts for learners when they design algorithms or study in any programming courses.

The result of the evaluation of the content was at the good level (4.21) which indicates that the content is related to construction and can improve logical thinking skills. It is also a pre-requisite for software developers. Results of the evaluation of outcome 3 (arithmetic expression) and outcome 4 (arithmetic and logical comparison) were very good (4.50 and 4.67) showing that these two outcomes are very important and related to logical thinking skills. The result of the evaluation of outcome 1 (general problem) and outcome 2 (probability problem) was only good (3.72 and 3.94) which implies that these two outcomes may not be directly related to logical thinking skills.

The instructional process for the development of logical thinking for software developers comprised the four steps of understanding, planning, executing and evaluation. The result of the evaluation of the instructional process was at the good level (4.36). The two steps of instructional process (understanding and planning) were evaluated at the very good level (4.50 and 4.61). The other two steps, executing and evaluation were evaluated at the good level (4.28 and 4.06).

The evaluation of the understanding step was at the very good level and emphasized that learners understand the problem by translating or restating the sentence of the problem. This step aligns with Ku *et al.* (2007) stated that the national assessment of educational progress that mathematical word problems are considered by students to be difficult at all age levels in elementary and secondary schools. Moreover, Bernardo, Mayer and Rosen stated that a major cause of the difficulty appears to be the student's ability to understand the problem structure that is embedded in the problem text.

The planning step was evaluated at a very good level that focuses on planning to solve the problem after the understanding step. This finding is similar to the study by Cifarelli (1998) that showed the relationship between cognitive activity and higher-level mathematical activity. The solver's cognitive acts of interpreting new situations in terms of their own conceptual structures and the ways that they resolved problematic situations that they faced along the way had a powerful influence on their subsequent solution activity.

The last two steps, execution and evaluation were evaluated only at the good level (4.28 and 4.06) whereas the 1st two steps, understanding and planning were evaluated at the very good level (4.50 and 4.61). This implies that the experts placed more emphasis on the 1st two steps than on the last two steps and indicates that in the experts' view if learners understand and plan very well it easy to execute and evaluate the result.

Moreover, if the overall evaluation of the instructional process is only at the good level, it means the instructional process is effective for the instruction of logical thinking skills for software developers but it is not the best instructional process. This study was limited in that it focused on software developers only in Thailand and it investigated the perspective of only twenty people for the content and only eighteen experts for the evaluation of the content and the instructional process. Future studies might be conducted in other settings and with more participants, especially in order to bring together both content and instructional processes to implement the development of logical thinking skills.

CONCLUSION

Logical thinking skills are considered to be important for learners in various fields (Greeno, 1989; Yaman, 2005; Kawamoto and Arai, 2008; Booneka and Kiattikomol, 2008). The results of this study indicate that the logical thinking skills of software developers may be divided into two parts, namely; the content and the instructional process. The content comprises four outcomes and similarly the instructional process comprises four steps. Together these two parts impact directly on the development of the learners' thinking ability.

Institutes and instructors could make use of the results of this study to develop curriculum with a focus on logical thinking skills for learners, especially for software developers but that are also applicable to other thinking skills. As instructors are conducting the course in logical thinking skills using the instructional process that the researchers have proposed, they may add outcomes to the content if it is believed that they will be of benefit the learners. Although, outside of the scope of the present research, future studies may focus both on the implementation of the content and instructional process of logical thinking skills for software developers and on the evaluation of the learners' performance.

ACKNOWLEDGEMENTS

Researchers wish to thank all participants in the study and anonymous people who facilitated while we did this research. Thank you to Dr. Elizabeth Murphy and Maria A. Rodriguez-Manzanares for their comments and revised on this study.

REFERENCES

Barnes, D.J., S. Fincher and S. Thompson, 1997. Introductory problem solving in computer science. Proceedings of the 5th Annual Conference on the Teaching of Computing, (ITiCSE'97), Centre for Teaching Computing, Dublin City University, Ireland, pp: 36-39.

Beyer, B.K., 2008. How to teach thinking skills in social studies and history. Soc. Stud., 99: 196-201.

Booneka, N. and P. Kiattikomol, 2008. Ranking competencies for software developer in Thailand. Proceedings of the EDU-COM International Conference on Sustainability in Higher Education: Directions for Change, Nov. 19-21, Edith Cowan University, Perth Western Australia, pp: 36-39.

Chang, K.E., Y.T. Sung and S.F. Lin, 2006. Computer-assisted learning for mathematical problem-solving. Compt. Educ., 46: 140-151.

Cifarelli, V.V., 1998. The development of mental representations as a problem solving activity. J. Mathematical Behav., 17: 239-264.

Creswell, J.W., 2008. Educational Research: Planning, Conducting and Evaluating Quantitative and Qualitative Research. 3rd Edn., Merrill/Prentice Hall, USA,, ISBN: 0136135501.

Greenes, C., 2009. Mathematics learning and knowing: A cognitive process. J. Educ., 189: 55-64.

Greeno, J.G., 1989. A perspective on thinking. Am. Psychol., 44: 134-141.

Kawamoto, K. and H.A. Arai, 2008. Evaluation of logical thinking ability through contributions in a learning community. Lecture Notes Comput. Sci., 4938: 326-333.

Ku, H.Y., C.A. Harter, P.L. Liu, L. Thompson and Y.C. Cheng, 2007. The effects of individually personalized computer-based instructional program on solving mathematics problems. Comput. Hum. Behav., 23: 1195-1210.

Marcou, A. and S. Lerman, 2006. Towards the development of a self-regulated mathematical problem solving. Prague PME, 4: 137-144.

Mayer, R.E., 2008. Learning and Instruction. 2nd Ed., Upper Saddle River, New Jersey.

NCTM, 1989. Curriculum and Evaluation Standards for School Mathematics. National Council of Teachers of Mathematics, Reston, VA.

Ralston, A., 2005. Do we need ANY mathematics in computer science curricula?. SIGCSE Bull., 37: 6-9.

Roadrangka, V., R.H. Yeany and M.J. Padilla, 1982. GALT, group test of logical thinking. University of Georgia, Athens, GA.

Slavin, R.E., 2007. Educational Research in an Age of Accountability. Pearson Education, Boston, MA.

Tobin, K.G. and W. Capie, 1981. The development and validation of a group test of logical thinking. Educ. Psychol. Measurement, 41: 413-423.

Verschaffel, L., E. de Corte, S. Lasure, G. van Vaerenbergh, H. Bogaerts and E. Ratinckx, 1999. Learning to solve mathematical application problems: A design experiment with fifth graders. Mathematical Thinking Learn., 1: 195-229.

Yaman, S., 2005. Exploring effectiveness of problem based learning on developing logical thinking skills in science teaching. J. Turk. Sci. Educ., 2: 31-33.