THE IDENTIFICATION AND VALIDATION OF THE ELEMENTS OF PROBLEM-SOLVING COMPETENCY OF PROGRAMMING TEACHERS BASED ON THE FUZZY DELPHI METHOD

Muhammad Modi Lakulu[1,*], Nor Masharah Husaini[2,*], Sulaiman Sarkawi[3]
Sultan Idris Education University, 35900, Perak, Malaysia
[*]For correspondence; modi@fskik.upsi.edu.my[1], masharah.husain@gmail.com[2]

ABSTRACT: *In everyday lives, people have to deal with a variety of problems, with each entailing a specific solution that relies on how they perceive such a predicament. Clearly, problem solving requires the ability to identify and to apply relevant skills systematically. Hence, the mastery of problem-solving skills is an important attribute of a competent teacher. Several techniques and strategies are available to help solve problems in computer science, especially in programming. As such, this study was carried out to determine the appropriate competency elements that programming teachers should have. A group of experts (consisting of 22 public school teachers and university lecturers) was recruited to elicit their expert opinions. Collected data were analyzed using the Fuzzy Delphi Method (FDM) to validate several elements of problem-solving competency. Through the analysis, all the competency elements were validated to be important based on a consensus of the experts, which exceeded the moderate level of agreements. The analysis also showed that the competency elements could be ranked in terms of their importance. Thus, with an appropriate programming teacher's competency model, consisting of such elements, efforts to develop competent programming teachers can be improved significantly. More importantly, with more competent programming teachers, student learning of programming can be further enhanced.*

Keywords: competency elements, programming, programming teacher, problem-solving competency

1. INTRODUCTION

Essentially, problems are obstacles faced by individuals as they strive to achieve certain goals in their everyday lives. To help understand and discuss the nature of problems, a number of scholars have proposed several definitions, such as [1] who defined a problem as a space that separates a current situation from the desired situation that entails a solution, containing a list of steps in solving such a problem. In fact, problem-solving has been recognized as on one of the important skills that everyone must have to deal with many challenges throughout his or her life. In short, problem-solving is a continual process or an activity in which we need to apply all the knowledge that we have in order to deal with unfamiliar, challenging phenomena or events.

Understandably, such a process involves skills and abilities that can help us to think intelligently in finding or determining the appropriate solution to a problem that we are faced with. In this regard, the main aim of education is helping students to learn in such a way that they can effectively apply all the knowledge that they have learned to solve any problem in any situation. As such, teachers are entrusted with the responsibility for developing and nurturing problem-solving skills of their students.

Problem-solving is the ability to identify and solve a problem using relevant skills systematically [2]. In essence, this ability necessitates appropriate actions to help solve a problem by making it more understandable, and therefore more solvable, such as using accurate algorithms more systematically. Clearly, problem-solving involves several cognitive processes to guide the formulation of solutions to problems [3]. Furthermore, [4] assert that problem-solving is a method that combines both cognitive and meta-cognitive process, which need to be emphasized in the teaching and learning process. Effectively, this amalgam of mental processes helps us to choose appropriate solutions to problems that we are facing with. Given the above prevailing issues, this study was carried out with the aim to explore elements of teachers' problem-solving competency in programming. As such, the main research question of the study is as follows: "What are the elements of problem-solving competency of programming teachers?"

2. LITERATURE REVIEW

Teachers as Problem Solver

In understanding the role played by teachers as problem-solver, [5] eloquently asserts that "the teacher should put himself [or herself] in the student's place, he [or she] should see the student's case, he [or she] should try to understand what is going on in the student's mind, and ask a question or indicate a step that could have occurred to the student himself." His assertion clearly underscores the importance of teachers to understand how their students think during the learning process. Of course, in such a process, teachers can use a variety of approaches or strategies to help stimulate students' reactions, such that the latter become active in the learning process.

Being cognitively stimulated, students will be able to focus on learning activities in which teachers can apply their knowledge and experiences in helping the former to solve programming problems. As students, naturally, their problem-solving ability is relatively inferior compared with their teachers', given the former's lack of experience or skills in conceptualizing and developing effective techniques or strategies in solving programming problems. Therefore, teachers have to play an important role in teaching and in guiding their students to acquire the right programming skills by which the latter can confidently deconstruct and analyze the logical components of a given problem that leads to a precise solution.

Obviously, the competency of teachers is assessed in terms of their ability in teaching a particular subject matter. Furthermore, teachers need to diversify their expertise and skills to help plan effective teaching activities in which their students can learn efficaciously. In this respect, Gagne (cited by Dwiyogo, [6]) emphasizes the important role of teachers in imparting knowledge of problem-solving to their students. Underscoring such a role entails teachers to be consciously aware of the most appropriate principles and strategies for solving problems that are specific to a particular field. Likewise, such an emphasis is shared by [7], who assert that teachers play a vital role in guiding and facilitating teaching and learning activities in the classroom.

Teaching Programming and Problem Solving

Undeniably, learning programming entails students to learn and apply various complicated techniques or methods. Thus, most students view the subject matter to be difficult to learn, let alone to master it. Further compounding the problem, virtually almost all programming topics contain many abstract concepts that are not only unfamiliar to students but also complicated. Thus, it is hardly surprising to observe programming students to be overwhelmed in learning how to solve a given problem, unless they have sufficient ability to think logically and critically. Unresolved, such pressing learning issues faced by students can result in poor achievement in programming. Hence, a basic knowledge and problem-solving skills are a prerequisite for successful learning [8].

In essence, the process of problem-solving entails a number of important aspects, including logical thinking, critical thinking, problem analysis, testing, and problem-solving [9]. Such an emphasis on such aspects is echoed by [10], who asserts that programming students need to have adequate logical thinking ability, which is an ability that supposedly has been developed when they were middle-grade students learning mathematics. Ironically, despite having such an ability, most students seem to struggle in applying their logical knowledge and skills in solving a given problem-solving task.

Moreover, [11] argue that the weakness of students in programming is indicative of their failure to use logical thinking in solving problems in their everyday lives. Thus, it is common for programming students to lament their hardships and difficulties in learning the subject matter. Likewise, [12] contend that the major problem in learning programming is attributed to weak problem-solving skills and poor logical thinking of students Such a contention is supported by a recent research finding of [13], who found students pursuing programming courses lacked the most important ability to learn to program, namely problem-solving skills and logical thinking. As such, problem-solving skills of programming students should be emphasized in learning to help them develop strong creative and critical minds. In this regard, a study by [14] found that training using real-world problems, decomposition analysis, planning, and specific domain knowledge could help improve students' problem-solving skills. To help realize such improvement, teachers should develop their teaching activities using appropriate problem-solving principles that would help them to teach complicated algorithms to their students more effectively. Furthermore, according to [15], students should be encouraged to solve problems related to programming constructs of program development to help them develop their problem-solving skills in programming. More importantly, at the end of a learning process, students should be able to apply their problem-solving skills in categorizing the main problem into several smaller problems such as to manage all these problems using a similar solution. Nonetheless, to achieve the above aims are not easy, given the acute lack of expertise and skills in problem-solving among teachers [16]. As such, learning becomes more student-centred, not teacher-centred with the teachers playing the role of a facilitator or a mentor.

From the teaching perspective, the research finding of [17] underscores the extent to which programming teachers' meta-cognitive and problem-solving skills will have an impact on student learning. They found that such teachers' skills provided opportunities for students to learn more efficaciously. As such, teachers' skills are crucial to the development of students' programming ability. As agreed by all scholars, producing quality students in programming field depends on quality teaching, with which programming teachers use their expertise and skills to the fullest. Clearly, teachers play an important role in the development of quality students by imparting appropriate knowledge and logical skills to students during programming activities.

Problem Solving Techniques

Often, a majority of people solve their everyday problems, which manifest in varying degrees of difficulty or complexity, without being consciously aware of the process involved. In contrast, in the teaching and learning process, students are usually given all the relevant information that they need, such as explicitly stated problems, required input, and expected output; thus they become fully aware of the process that is needed to solve such a given problem. However, in reality, such a process is not that easily discernible to help determine a specific solution. In such a case, students need to first identify the problem, then make a decision by using available information, and finally determine the expected outcome.

Arguably, applying a particular problem-solving method to help solve problems involves a learning process through which students need to think critically and decisively. In this regard, several methods and strategies can be used to produce a viable solution to a particular problem. However, accordingly, to [6], strategies used in solving problems in a particular domain may not be suitable to be applied to solving problems in other domains.

Nonetheless, irrespective of different problem domains, all problem-solving strategies must involve a set of implementation activities to help formulate a particular solution to a specific problem. Arguably, the successful implementation of such strategies relies on existing knowledge, skills, and experiences of individuals in dealing with similar problems[18].

The improvement in students' problem-solving skills can be reliably ascertained if they can demonstrate their ability to produce an effective strategy to solve a given problem [19]. Moreover, the performance of students during the problem-solving process is not static. In fact, it will change from being low at the beginning to high at the end of this process, thus reflecting the gradual progression of an incremental improvement in students' understanding [5]. To this end, several models have been proposed in previous studies to help students gain a sound knowledge of strategies for problem-solving. For example, problem-solving model is one of the models that have been studied that focus on problem-solving. Specifically, this model is based on a heuristic, which is a strategy that is commonly used to solve mathematical problems, which does not necessarily result in producing solutions. Essentially, [5] model consists of four phases, namely understanding the problem, planning the solution strategy, implementing the solution strategy, and evaluating and repeating the solution. Each phase of this model has to be executed carefully and systematically to ensure students will be able to improve their skills in making a decision, judgement, and logical evaluation.

In the programming field, choosing the appropriate approach to derive a solution to a given problem is not easy. Nonetheless, some problems can be solved using a set of fixed actions or steps called algorithmic solution. After having determined a unified procedure, solutions can be ascertained by implementing those steps in a complete sequence called an algorithm. In the problem-solving phase of programming, programmers need to design an algorithm, not to follow an algorithm. As such, programmers need to be fully aware of the appropriate strategy that is needed in solving programming problems.

In a more detailed proposal, [8] proposed six steps in problem-solving of programming, namely identifying a problem, understanding the problem, identifying solutions, selecting the best solution, preparing a set of solution steps, and evaluating the final solution. However, these steps are not rigid as they can change to help produce relevant solutions in algorithmic or in heuristic form. However, a solution formulated may be ineffective if programmers failed to identify and understand a problem at the early stage of the process. Moreover, such a solution formulated may not be precise or incomplete, which can result in an unexpected or unintended outcome.

On the other hand, [20] proposed three processes in problem-solving, namely analysis and synthesis, development of a common solution (algorithm), and validation of the algorithm. Additionally, in their study, they implemented the process of problem-solving by translating the algorithm to a programming language (program) and testing the program to determine whether the solution was able to solve a given problem. After the program had been composed, it then entered the third phase, which was the maintenance phase. This phase involved program usability that was performed to check whether the program could either be maintained as it was or be modified further to meet changes in requirements or to correct mistakes that emerged during its use.

In general, the problem-solving process in programming can be viewed as a process that analyses a given problem, divides the problem into several smaller problems (sub-problems), and develops a common solution, called an algorithm, for each sub-problem. Finally, the solutions to all sub-problems are added together to form a program that helps solve the main (actual) problem. Nonetheless, the process of understanding and analysing problems will normally take more time than what is usually anticipated by programmers. Of course, a problem-solving process in programming can become more complex, entailing longer time to complete, if such a process starts on a wrong foot.

3. RESEARCH METHODOLOGY

The main aim of this study was to identify the elements of problem-solving competency, which will serve as guidelines for the development of quality programming teachers of secondary schools in Malaysia. To help validate such elements, the researcher carried out a validation process through which the elements were reviewed and analysed for their importance. This process involved the analysis on a list of the competency elements, which were conceptualized and identified from the preliminary findings of a needs analysis survey and from the findings reported in the literature. The output of this analysis was a list of competency elements, which was submitted to a panel of experts in the first round of the validation process. For this purpose, 22 experts comprising practitioners in the field of programming pedagogy and programming language were selected.

The research method used for this study was based on the Fuzzy Delphi Method, which was originally proposed by [21] and later revised by [22]. This method is used to obtain quick and precise validation of factors or elements based on a consensus of experts in a single round [23]. Furthermore, the strength of the FDM lies in its ability to produce consistent validation results [24], and it is efficient as it does not require many rounds of validation. For this study, the researcher engaged a group of experts, consisting of 22 expert teachers and university lecturers who had vast experience in teaching computer science and programing, as the respondents. Such a number of experts is consistent with the recommended number of 10 – 50 respondents, as suggested by [25].

4. DATA ANALYSIS AND RESULTS

To reliably apply the FDM, researchers have to fulfil two requirements. First, the values of the triangular fuzzy number need to have three scales that are translated from the Likert scales. In this study, 5-point Likert scales were used to measure the level of agreements of respondents as follows: "1" for strongly disagree, "2" for disagree, "3" for moderately agree, "4" for agree, and "5" for strongly agree. The triangular fuzzy number has three values, namely *m1*, *m2*, and *m3* that refer to the lowest, reasonable, and highest value, respectively. The triangular fuzzy number must produce a threshold value of less than 0.2 [26][27]. For the second requirement, the percentage of agreement of experts calculated by the defuzzification process must exceed 75%. Table 1 summarizes the defuzzification value of each element of the problem-solving competency, and all elements were ranked based on their levels of importance.

As shown, all the three elements of problem-solving competency fulfilled both requirements of the FDM, which was made evident by the threshold value (d) of less than 0.2 and the percentage of agreement of experts exceeding 75%. In addition, the findings showed that the definition of a programming problem was rated as the most important element, followed by the analysis of the problem. Finally, the preparation of an appropriate solution to the problem based on the level of students' knowledge was rated as the third most important element.

Table 1. The defuzzification value of each element of problem-solving competency

	Element of competency	Threshold (d) Values	Defuzzification value				Percentage of agreement	Ranking	Expert consensus
			m1	*m2*	*m3*	Fuzzy (A)			
1	Defining a programming problem.	0.12	0.54	0.74	0.94	0.74	100 %	1	Accept
2	Analyzing the programming problem.	0.05	0.58	0.78	0.98	0.78	95 %	2	Accept
3	Preparing a solution to the problem based on the level of students' knowledge.	0.14	0.53	0.73	0.93	0.73	95 %	3	Accept

5. DISCUSSIONS

The findings of this study suggest that, to be highly competent, programming teachers must have three elements of problem-solving competency. Firstly, they should be able to define a programming problem holistically such that they can help stimulate the thinking process of students to deal with the given problem. Additionally, teachers need to guide students in developing a sound understanding of the problem by relating it with relevant information. Secondly, teachers should emphasise the analysis of the problem by focussing on relevant strategies that can help students solve the problem. Achieving this relies on teachers' ability to decompose the main problem into smaller sub-problems that are manageable for students to solve. Thirdly, teachers must have the ability and skill to process all relevant solutions that have been formulated earlier.

Moreover, teachers should also provide students with appropriate concepts and information of relevant problem-solving techniques that are appropriate for students based on their current level of knowledge. Such a process entails reviewing the impact of the solution method used and, if required, reusing the method with a new set of procedures. The validation process performed in this study showed that all the elements of problem-solving competency received high ratings, surpassing the moderate level of agreement among the experts. Likewise, the same process showed that such elements received different rankings, suggesting they differed in terms of importance.

6. CONCLUSION

Evidently, programming teachers must possess strong thinking skills, notably in interpreting and analysing problems and in deciding the appropriate problem-solving techniques to use, thus helping them to teach their students more efficaciously. Effectively, the problem-solving process helps strengthen students' critical minds and provides opportunities for teachers to gauge their students' ability in overcoming problems. In fact, the problem-solving steps discussed in this paper can be transferred to solving everyday problems, in particular to solving problems related to programming. Moreover, solution techniques that can deconstruct problems logically are both efficient and effective.

In light of the above findings, the emphasis on teachers' problem-solving skills cannot be overstated, as they will have a huge impact on student learning, such that students can become critical and creative as they learn to become good programmers. Therefore, programming teachers need to master all the elements of problem-solving competency to help improve the current teaching practice in Malaysia.

7. REFERENCES

[1] Hayes, J. R. (1981). *The Complete Problem Solver*. *Franklin Institute Press* (2nd ed.). Philadelphia: Franklin Institute Press.

[2] Cohen, H., & Stemmer, B. (2007). *Consciousness and Cognition: Fragments of Mind and Brain*. *Elsevier*. United State: Academic Press. https://doi.org/10.1016/B978-0-12-373734-2.50001-7

[3] Wang, Y., & Chiew, V. (2010). On the cognitive process of human problem-solving. *Cognitive Systems Research*, *11*(1), 81–92. https://doi.org/10.1016/j.cogsys.2008.08.003

[4] Neufeld, V. R., & Barrows, H. S. (1974). The "McMaster Philosophy": an approach to medical education. *Journal of Medical Education*, *49*(1), 1040–1050. Retrieved from https://eric.ed.gov/?id=EJ106076

[5] Polya, G. (1957). *How to Solve It : A new aspect of mathematical method*. *The Mathematical Gazette*. Princeton, New Jersey: Princeton University Press. https://doi.org/10.2307/3609122

[6] Dwiyogo, W. D. (2016). Teaching and Learning Process : Thinking and Problem Solving. *The Online Journal of New Horizons in Education*, *6*(3), 121–129.

[7] Govender, I., & Govender, D. W. (2012). A constructivist approach to a programming course: Students' responses to the use of a Learning Management System. *African Journal of Research in Mathematics, Science and Technology Education*, *16*(2), 238–252. https://doi.org/10.1080/10288457.2012.10740742

[8] Sprankle, M., & Hubbard, J. (2012). *Problem Solving and Programming Concepts* (9th ed.). Pearson.

[9] Saeli, M., Perrenet, J., Jochems, W. M. G., & Zwaneveld, B. (2011). Teaching Programming in Secondary School : A Pedagogical Content Knowledge Perspective. *Informatics in Education*, *10*(1), 73–88.

[10] Kordaki, M. (2010). A drawing and multi-representational computer environment for beginners' learning of programming using C: Design and pilot formative evaluation. *Computers and Education, 54*(1), 69–87. https://doi.org/10.1016/j.compedu.2009.07.012

[11] Butler, M., & Morgan, M. (2007). Learning challenges faced by novice programming students studying high level and low feedback concepts. In *Proceedings of the 24th ascilite Conference* (pp. 99–107). Singapore.

[12] Gomes, A., & Mendes, A. J. (2007). An environment to improve programming education. In *International Conference on Computer Systems and Technologies -*

CompSysTech (pp. 1–6). Rousse, Bulgaria.

[13] Faradilla, T., Mujaiyid, Z., Baidowi, A., & Jaafar, J. (2013). Learning Introductory C Programming : Relevant Exercises Based On Student Difficulties Factors. In *International and National Conference on Engineering Education (INCEE '11)* (pp. 1–6). Phuket.

[14] Pea, R. O. Y. D., & Kurland, D. M. (1984). On The Cognitive Effects Of Learning Computer Programming. *New Ideas Psychology*, *2*(2), 137–168.

[15] Kumar, A. N. (2003). Learning programming by solving problems. In *Informatics Curricula and Teaching Methods* (pp. 29–39). https://doi.org/10.1007/978-0-387-35619-8_4

[16] Dunican, E. (2002). Making The Analogy : Alternative Delivery Techniques for First Year Programming Courses. In *14th Workshop of the Psychology of Programming Interest Group* (pp. 89–99). London.

[17] Havenga, M., & Breed, B. (2013). Metacognitive and Problem-Solving Skills to Promote Self-Directed Learning in Computer Programming : Teachers ' Experiences. *SA-eDUC JOURNAL*, *10*(2), 1–14.

[18] Ismail, S., & Atan, A. (2011). Aplikasi Pendekatan Penyelesaian Masalah Dalam pengajaran Mata Pelajaran Teknikal dan Vokasional di Fakulti Pendidikan UTM. *Journal of Educational Psychology and Counseling*, *2*, 113–144.

[19] Kantowski, M. G. (2010). Processes Involved in Mathematical Problem Solving. *Journal for Research in Mathematics Education*, *8*(3), 163–180. https://doi.org/10.2307/748518

[20] Dale, N., McMillan, M., Weems, C., & Headington, M. R. (2002). *Overview of Programming and Computational Theory* (1st ed.). Jones & Bartlett Learning.

[21] Murray, T. J., Pipino, L. L., & Gigvh, J. P. van. (1985). A pilot study of fuzzy set modification of Delphi. *Human Systems Management*, *5*(1), 76–80. Retrieved from http://content.iospress.com/articles/human-systems-management/hsm5-1-11

[22] Kaufmann, A., & Gupta, M. M. (1988). *Fuzzy mathematical models in engineering & management science*. *Elsevier Science Inc*. New York: Elsevier Science Publisher. https://doi.org/10.1016/0377-2217(89)90024-6

[23] Mohd Ridhuan, M. J., Saedah, S., Farazila, Y., Nurulrabihah, M. N., Zaharah, H., & Ahmad Ariffin, S. (2015). Aplikasi Teknik Fuzzy Delphi Terhadap Keperluan Elemen Keusahawanan Bagi Pensyarah Kejuruteraan Politeknik Malaysia. *International Journal of Business and Technopreneurship*, *5*(1), 135–150.

[24] Habibi, A., Sarafrazi, A., & Izadyar, S. (2014). Delphi Technique Theoretical Framework in Qualitative Research. *The International Journal Of Engineering And Science*, 2319–1813.

[25] Jones, H., & Twiss, B. C. (1978). *Forecasting Technology for Planning Decisions*. London: Palgrave Macmillan NY. https://doi.org/10.1007/978-1-349-03134-4

[26] Cheng, C.-H., & Lin, Y. (2002). Evaluating the best main battle tank using fuzzy decision theory. *European Journal of Operational Research*, *142*, 174–186. https://doi.org/10.1016/S0377-2217(01)00280-6

[27] Chu, H. C., & Hwang, G. J. (2008). A Delphi-based approach to developing expert systems with the cooperation of multiple experts. *Expert Systems with Applications*, *34*(4), 2826–2840. https://doi.org/10.1016/j.eswa.2007.05.034