

Components and Indicators of Problem-solving Skills in Robot Programming Activities

Chacharin Lertyosbordin, Sorakrich Maneewan, Daruwan Srikaew

Faculty of Industrial Education and Technology
King Mongkut's University of Technology
Thonburi Bangkok, Thailand

Abstract—The objective of this research was to study the components and indicators of problem-solving skills in robot programming activities for high school students. This is done by analyzing the second order of confirmatory factor analysis (CFA) based on data from the behavioral assessment with regard to the robot programming activities of 320 students from specialized science schools. The results of the research revealed that the problem-solving skills in robot programming activities had five components and 15 indicators. All the components were tested for consistency using CFA statistics with the support of R-Studio program. The model analysis results were found to be consistent with empirical data with Chi-Square = 98.273, df = 80.000, p-value = 0.081, GFI = 0.961, NFI (TLI) = 0.924, CFI = 0.985, RMSEA = 0.027, RMR = 0.007. This indicates that all the identified components and indicators are involved in problem-solving skills in the robot programming activities of high school students.

Keywords—Components; indicators; problem solving; robot programming

I. INTRODUCTION

The Organization for Economic Cooperation and Development (OECD) [1] has published the OECD Future of Education and Skills 2030 report, indicating that robotics engineering is the number one task that the world will need. Consequently, inspiring students and encouraging learning with regard to robotics engineering should go along with the critical thinking and problem-solving skills [2] that are components of learning and innovation within the 21st Century Skills [3]. These are widely known in education, and are in line with the views of the World Economic Forum (2016) [4] which has defined critical thinking and problem-solving skills as the core competencies for students in 21st Century. In addition, both of these higher-order thinking skills are essential aspects of fostering skills across all learning and innovation skills groups [5].

In this research we focus on problem-solving skills. By this we mean the capability to use thinking methods based on knowledge and experience, in order to achieve the expected goals. This is done in a step-by-step fashion, by gathering and linking factors and facts [6 - 8]. These skills are considered to be the most important and fundamental skills for learners in the 21st Century [3,4]. Therefore, at present, the education profession is making great efforts to devise a learning management approach that can be used to improve problem-

solving skills. Since 2018, the Ministry of Education of Thailand [9] has set the standard for developing thinking skills, calculation skills, analytical thinking skills and systematic problem-solving skills at the high school level. Moreover, the Ministry of Science of Thailand [10] has encouraged the organization of robot programming activities for high school students in order to support high school educational standards throughout the country.

Robot programming is an activity used as part of the learning process for the development of many important 21st Century skills [11 - 14], especially problem-solving, which are proven to respond well to programming [15 - 18]. At the present time, Thailand has no clear standards for assessing problem-solving skills in terms of robotics programming activities for high school students [19]. In addition, the researchers found that the search term "problem solving skill component and indicator on robot programming" does not appear in any Google Scholar databases from 2010 to 2020, nor are there any concrete elements and indicators. Therefore, this research studied the components and indicators of problem-solving skills in programming activities for high school students in Thailand to model prototype guidelines that can be used for student activity design and skill measurement in the future.

II. RESEARCH OBJECTIVE

- 1) To study the components and indicators of problem-solving skills in robot programming activities for high school students.
- 2) To design the model hypothesis of components and indicators of problem-solving skills in robot programming activities for high school students.
- 3) To evaluate the validity of the model hypothesis of components and indicators of problem-solving skills in robot programming activities for high school students by analyzing the second order of confirmatory factor analysis.

III. THEORETICAL FRAMEWORK

In conducting this research, the researcher reviewed the literature to synthesize the components and indicators of problem-solving skills in robot programming activities for high school students. This aspect is divided into two main parts: Part I, problem solving skills; Part II, robot programming procedures. The details are as follows:

A. Part I: Problem-solving Skills

Problem-solving skills refer to a capability to use thinking methods based on knowledge and experience to achieve an expected goal. This is done step-by-step, by gathering and linking factors and facts [6 - 8]. Consequently, many scholars have detailed the components of problem-solving skills including Bransford and Stein [20] who defined the problem-solving skill component as “IDEAL” which consists of 1) identify 2) define 3) explore 4) act and 5) look back. This conforms with the work of Foshay and Kirkley [21] who defining the solution components in terms of principles for teaching problem solving to be consistent with 1) identifying the problem 2) defining the problem through thinking about it and sorting out the relevant information 3) exploring solutions through looking at alternatives, brainstorming, and checking out different points of view 4) acting on the strategies 5) looking back and evaluating the effects of the activity. In addition, there are Polya's problem solving techniques [23] that is a generally-accepted problem-solving process in mathematics which consists of four steps to solve a problem. These are as follows: 1) understand the problem 2) devise a plan (translate) 3) carry out the plan (solve) 4) look back (check and interpret). The details with regard to the components of problem-solving skills from many other sources can be synthesized as shown in Table I.

From Table I, which is the result of the synthesis of the components of problem-solving skills gleaned from theories and academic articles, the researcher can conclude that the components of problem-solving skills consist of five elements:

1) *Identifying the problem.* This refers to explaining the details and the boundaries of the problem, and determining what needs to be solved.

2) *Goal setting.* This refers to sorting out related information which leads to things which need to be done for the problem to be resolved.

3) *Creating a solution.* This refers to looking for alternative ways to resolve a particular problem through brainstorming and reviewing to create a problem solution.

4) *Acting on the solution.* This refers to the implementation of the created solution.

5) *Returning to check the results.* This refers to the assessment of the results, and evaluating the success in order to ensure the problem has been solved.

B. Part II: Robot Programming Procedures

Robot programming refers to the control of a robot by writing computer programs which can be used to create and instruct the mechanical device using electronic systems to perform a desired task [26 - 29]. Nowadays, many scholars have detailed the procedures of programming. For example, Sharma [30] described the Software Development Life Cycle (SDLC) as a process that programmers use to create productivity outcomes. These consist of 1) requirement analysis 2) planning 3) software design 4) software development 5) testing and 6) deployment. These conform to the five-step programming process outlined in Wikibooks [32]: 1) clarifying / defining the problem 2) designing the program, 3) coding the program, 4) testing the program, and 5) documenting and maintaining. In addition, the School of Computer Science at the University of Birmingham [36] provides a simple four step programming procedure: 1) identify the problem 2) design a solution 3) write the program and 4) check the solution. The detail of the programming procedures from many other sources can be synthesized as shown in Table II.

TABLE I. SYNTHESIZATION OF PROBLEM-SOLVING SKILLS

Problem-solving's Components	IDEAL Aspects [20]	PLATO Learning Aspects [21]	Jonassen [8]	Great Schools Partnership Aspects [22]	POLYA Aspects [23]	Astuti, Suranto & Masykuris [24]	Franestian, Suyanta & Wiyono [25]
Identifying the problem	✓	✓	✓	✓	✓	✓	✓
Goal setting	✓	✓	✓	✓	✓	✓	✓
Creating a solution	✓	✓	✓	✓	✓	✓	✓
Acting on solution	✓	✓	✓	✓	✓	✓	✓
Returning to check results	✓	✓	✓	✓	✓	✓	✓

TABLE II. SYNTHESIZATION OF ROBOT PROGRAMMING PROCEDURES

Robot programing Procedure	Sharma [30]	Valenzuela [31]	Wikibooks [32]	Amjo [33]	Person [34]	Department of Computer Science and Statistics The University of Rhode Island [35]	School of Computer Science University of Birmingham [36]
Identify the Problem	✓	✓	✓	✓	✓	✓	✓
Design a Solution	✓	✓	✓	✓	✓	✓	✓
Code the Program	✓	✓	✓	✓	✓	✓	✓
Test the Program	✓	✓	✓	✓	✓	✓	✓
Implement the Program	✓	✓	✓	✓	✓	✓	✓

From Table II, the researcher can conclude that synthesizing the robot programming procedures consists of five steps:

1) *Identifying the problem.* This refers to analyzing the problem and determining the Input and Output processes that need to be incorporated in order to solve the problem.

2) *Designing a Solution.* This refers to arranging the order of the algorithms used by drawing flowcharts or writing pseudo code.

3) *Coding the program.* This refers to converting the instructions and sequence of methods from the flowchart to computer language.

4) *Testing the program.* This refers to the validation of the grammar of the computer language and the interpretation of the results for computer operation purposes. In addition, it involves testing for compatibility with hardware, including the Input and Output Processes.

5) *Implementing the program.* This refers to the results based on the program used. This should be followed up with further improvements.

From studying the problem-solving skill components and the robot programming procedures as shown above, the researcher can create a conceptual framework with regard to problem-solving skill components, and the indicators in robot programming activities. This framework is as shown in Fig. 1.

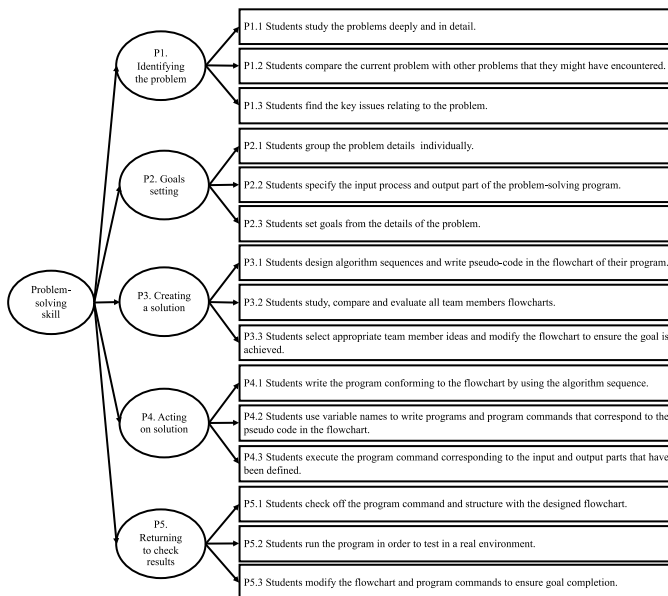


Fig. 1. Conceptual Framework with Regard to Problem-Solving Skill Components and the Indicators in Robot Programming Activities (Model Hypothesis).

IV. RESEARCH METHODOLOGY

A. Population

In this research, the population consisted of students in a group of the Princess Chulabhorn's Science High Schools (one of the Thai governments specialized scientific schools) who had completed a course in "robot programming" in the academic year 2020. This involved three schools in the northern province, three schools in the central province, three schools in the northeastern province, and three schools in the southern province; a total are 12 schools; each of which had 40 students in grade 10, and 40 students in grade 11, a total are 80 students per school. Consequently, the population was 960.

B. Samples

The researcher performed multi-stage sampling, starting with a sampling unit of three schools which was randomized to two schools per province, for a total of eight schools and 640 students. Then, stratified random sampling was used to select 30 students in grade 10 and 30 students in grade 11. As a result, the sample group was 60 students per school, totaling 480 students. The final step was taking a random sample of 40 students per school, and a total net sample of 320 students. This number conformed to the sampling size suggested by Yamane [37] and related to the appropriate sample size for structural equation model (SEM) statistics [38].

C. Research Instrument

In this research, the researcher used behavioral frequency self-assessment in terms of robot programming activities which was validated by seven experts. The assessment design was based on the Likert 5-point Scale to divide the behavior frequency (Level 5 = Always, 4 = Usually, 3 = About half the time, 2 = Seldom and 1 = Never).

D. Research Experts

To validate the research instrument in the form of a behavioral frequency self-assessment instrument in terms of robot programming activities, the researcher worked with seven experts from various fields, whose qualifications were as follows:

- 1) Two Lecturers in educational evaluation.
- 2) Two Lecturers in Computer engineering.
- 3) One Lecturer in Educational technology.
- 4) One Lecturer in Psychology.
- 5) One Psychiatrist with at least 5 years of adolescent behavior research experience.

V. PROCEDURES

The research was conducted in five steps as follows:

1) Study of the theories and research related to “Problem-solving skills” and “Robot programming” to identify the components and indicators of problem-solving skills with regard to robotics programming activities to obtain model hypotheses.

2) Creation of behavioral assessment in order to identify problem-solving skills in terms of the robot programming activities. This was based on the components and indicators of problem-solving skills in robot programming activities that followed the researcher’s model hypotheses.

3) Evaluation of the validity of the behavior assessment instrument by using the Index of Item-Objective Congruence (IOC) involving seven experts.

4) Collection of behavior assessment data (Sum of the sample made up of 320 students in eight schools.).

5) Analyzing the data using confirmatory factor analysis (CFA) with the support of R-Studio software to examine the construct validity of the problem-solving skill components and indicators with regard to robot programming activities.

VI. RESEARCH RESULTS

1) To create and evaluate the validity of the behavioral frequency assessment instrument, seven experts determined

the validity of the questionnaire in robot programming activities for high school students using the Index of Item-Objective Congruence (IOC). The results had been shown in the Table III conclude that the statements in the model hypothesis are acceptable.[39].

2) From Table IV, it can be seen that the 5 components have a standardized factor loading (β) greater than 0.4, and consequently passed the standard statistical criteria [40]. In addition, by examining the goodness of it statistical indicator, the Chi-Square value was 98.27 at 80.00 degrees of freedom (df), with a probability (p-value) of 0.08. In addition, Goodness of Fit Index (GFI), Adjusted Goodness of Fit Index (AGFI), Normed Fit Index (NFI) and Comparative Fit Index (CFI) were 0.96, 0.94, 0.92 and 0.99, respectively. The Standardized Root Mean Square Residual (SRMR) was 0.04 and the Root Mean Square Error of Approximation (RMSEA) was 0.03. It can be concluded that all 15 components and 5 indicators with regard to problem-solving skills in the robot programming activities fit the empirical data, and correspond to the researcher’s hypothesis [41].

3) From Table V and Fig. 2, it can be seen that 15 indicators have a standardized factor loading (β) greater than 0.4. Consequently, they achieved the standard statistical criteria [40]. As a result, these 15 indicators could be appropriate for use as element of all five components.

TABLE III. SYNTHETIZATION OF PROBLEM-SOLVING SKILLS EVALUATION OF THE VALIDITY OF THE QUESTIONS IN THE BEHAVIORAL FREQUENCY ASSESSMENT

Robot programming procedure	Problem – solving skills		IOC	Meaning
	Component	Indicator		
Identify the problem	P1. Identifying the problem	P1.1 Students study the problems deeply and in detail	1.00	Accept
		P1.2 Students compare the current problem with other problems that they might have encountered	1.00	Accept
		P1.3 Students find the key issues relating to the problem	1.00	Accept
	P2. Goals setting	P2.1 Students group the problem details individually	1.00	Accept
		P2.2 Students specify the input process and output part of the problem-solving program.	1.00	Accept
		P2.3 Students set goals from the details of the problem.	1.00	Accept
Design a solution	P3. Creating a solution	P3.1 Students design algorithm sequences and write pseudo-code in the flowchart of their program.	1.00	Accept
		P3.2 Students study, compare and evaluate all team members flowcharts.	1.00	Accept
		P3.3 Students select appropriate team member ideas and modify the flowchart to ensure the goal is achieved.	1.00	Accept
Code the program	P4. Acting on solution	P4.1 Students write the program conforming to the flowchart by using the algorithm sequence.	1.00	Accept
		P4.2 Students use variable names to write programs and program commands that correspond to the pseudo code in the flowchart.	1.00	Accept
		P4.3 Students execute the program command corresponding to the input and output parts that have been defined.	1.00	Accept
Test the program / Program implementation	P5. Returning to check results	P5.1 Students check off the program command and structure with the designed flowchart.	1.00	Accept
		P5.2 Students run the program in order to test in a real environment.	1.00	Accept
		P5.3 Students modify the flowchart and program commands to ensure goal completion.	1.00	Accept

TABLE IV. FACTOR LOADING OF THE INDICATORS FOR EACH COMPONENT

Components	β_i	b_i	S.E.	R ²
P1	0.88	1.00	-	0.77
P2	0.86	0.94	0.13	0.75
P3	0.94	0.82	0.12	0.88
P4	0.56	0.52	0.08	0.32
P5	0.61	0.46	0.08	0.38

Chi-Square = 98.27, df = 80.00, p-value = 0.08, GFI = 0.96, AGFI = 0.94, NFI = 0.92, CFI = 0.99, SRMR = 0.04, RMSEA = 0.03

TABLE V. FACTOR LOADING OF PROBLEM-SOLVING SKILL COMPONENTS IN ROBOT PROGRAMMING ACTIVITIES

Com.	P1			P2			P3			P4			P5			R ²
	β_i	b_i	S.E.	β_i	b_i	S.E.	β_i	b_i	S.E.	β_i	b_i	S.E.	β_i	b_i	S.E.	
P1.1	0.67	1.00	-													0.45
P1.2	0.63	0.92	0.10													0.36
P1.3	0.67	0.99	0.11													0.44
P2.1				0.65	1.00	-										0.41
P2.2				0.57	0.86	0.11										0.33
P2.3				0.49	0.79	0.12										0.24
P3.1							0.51	1.00	-							0.26
P3.2							0.59	1.13	0.17							0.35
P3.3							0.65	1.23	0.15							0.42
P4.1										0.71	1.00	-				0.50
P4.2										0.61	1.00	0.10				0.38

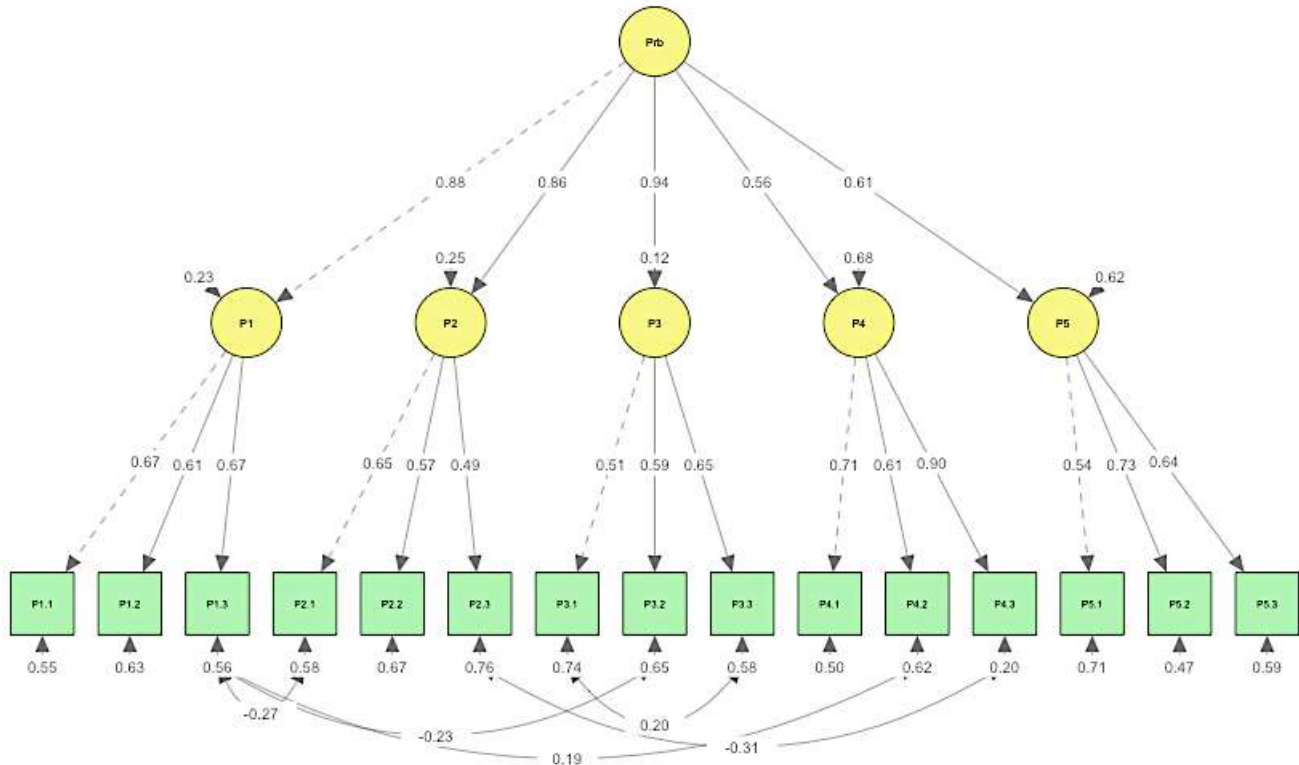


Fig. 2. Factor Loading Diagram of Components and Indicators of Problem-solving Skill in Robot Programming Activities on the Part of High School Students.

VII. CONCLUSION

From the results, the researcher can summarize that the components and indicators with regard to problem-solving skills in robot programming activities for high school students, are composed of five components and fifteen indicators. The details of the supporting information are as follows:

1) Component P3 (Creating a solution) had the highest factor loading at 0.94, followed by component P1 (Identifying the problem) which had a loading factor of 0.88. The third component was P2 (Goals setting) which had a loading factor of 0.86. In addition, the factor loading of the 3 components was more than 0.7, implying to these 3 components have sufficiency variance effected on problem-solving skill [42]. Moreover, the conclusion confirms to the meaning of the phrase "Problem-solving-approach" as defined by the APA Dictionary of Psychology [43]. That is, it is "The process whereby difficulties, obstacles, or stressful events are addressed through the use of coping strategies." Accordance to Jonassen [8], programming activities could be classified as one solution for design problem-solving that focuses on analysis and planning. This also corresponds to Chandrasekaran [44], who states that the key to problem-solving is the critical thinking step, in order for the student to understand the problem, and determine the structure and sequence of work to fit the problem.

2) The factor loading of component P4 (Acting on the solution) and component P5 (Returning to check results) were 0.56 and 0.61, respectively. This means that both of these components are important when it comes to problem-solving skills and is statistically acceptable [40]. Similarly, McFadden [45] demonstrated that programming must follow the plan strictly in order to achieve its goals. In addition, the continuous review and development of the results can increase problem-solving fluency. In line with Taheri, Sasaki and Ngetha [46], we can conclude that problem-solving skills can be built from the programming process, while repeating trials are also a part of the development of problem-solving ability. Moreover, Glenda and Westhuizen [47], also suggest that programming can demonstrate logical reasoning, numerical reasoning and language reasoning, all of which are essential components of problem-solving skills.

3) Indicator P1.1 (Study the problems deeply and in detail), P1.2 (Compare the current problem with other problems that they might have encountered), and P1.3 (Find the key issues relating to the problem) have factor loadings of 0.67, 0.63 and 0.67, respectively. These three indicators are statistically acceptable, which means that all the indicators under component P1 (Identifying the problem) are elementary problem-solving skills [40]. According to Kember [48], identifying the problem means the perception of the problem, and the development of a clear problem framework from digesting information. This leads to effective and accurate problem identification. This in line with Staniewicz [49] who said that problem identification is the first step in coming up with an engineering solution by gathering previously used

similar concepts to those of the present problem, then identifying what should be improved.

4) Indicator P2.1 (Group the problem details individually), P2.2 (Specifying the input process and the output of the problem-solving program), and P2.3 (Set goals from the details of the problem) have factor loadings of 0.65, 0.67 and 0.49, respectively. These three indicators are statistically acceptable, which means that all the indicators under component P2 (Goals setting) are elementary problem-solving skills [40]. This conforms with Wharton Executive Education [50], which indicates that a good solution should not stick by a lot of problem information, but should be manage with factor-and-effect relationships to set a new definitive scope for the problem. Moreover, Markman [51] explained that problem resolution targeting involves classifying a large number of existing problems and looking for a correlation in terms of the information to achieve inventive and precise solutions.

5) Indicator P3.1 (Designing algorithm sequences and write pseudo-code in the flowchart of their program), P3.2 (Study, compare and evaluate all team members' flowcharts), and P3.3 (Select appropriate team members' ideas and modify the flowchart to achieve the goal.) have factor loadings of 0.51, 0.59 and 0.65, respectively. The fact that these three indicators are statistically acceptable, means that all the indicators under component P3 (Creating a solution) are elementary problem-solving skills [40]. According to Crews & Ziegler [52], writing flowcharts with regard to algorithmic sequencing and pseudo-coding was a process of great importance with regard to engineering solutions. Additionally, Bryant [53] encouraged the use of the Critique, Explore, Compare, and Adapt framework in the process of designing, in that it can give results that are both comprehensive and well-informed.

6) Indicator P4.1 (Write the program conforming to the flowchart by using the algorithm sequence), P4.2 (Use variable names to write programs and program commands that correspond to the pseudo code in the flowchart.), and P4.3 (Execute the program command corresponding to the input and output parts that have been defined) have factor loadings of 0.71, 0.61 and 0.90, respectively. That these three indicators are statistically acceptable means that all the indicators under component P4 (Acting on a solution) are elementary of problem-solving skills [40]. According to Whipp, Tenkanen and Heikinheimo [54], computer programming is a step-by-step aspect of the problem-solving process. Naming each variable and choosing the correct command will support the solution of the problem more easily. It also conforms with Bilotta and Pantano [55], who explained that programming must consider the number and type of sensors which are part of the input, as well as the number and size of motors that are part of the output, to achieve the robot mission satisfactorily.

7) Indicator P5.1 (Check off the program command and structure with the designed flowchart), P5.2 (Run the program by testing in a real environment), and P5.3 (Modify the

flowchart and program commands to achieve the goal completely) have factor loadings of 0.54, 0.73 and 0.64, respectively. That these three indicators are statistically acceptable means that all the indicators under component P5 (Returning to check results) are elementary of problem-solving skills [40]. This conforms with Young, Sharlin and Igarashi [56], who explained that robot control programming needs validation results in terms of the robot's performance in a real environment. The program developer will systematically collect data with regard to the robot's performance in order to improve the robot operation in order to complete the mission. In addition, when solving a robot foot walking problem, Maicon, Aramizo, Houman and Mohammad [57] found that using a robot errors recovery programming technique involving equation generation could reduce the errors in the next operation, and solve the problem more quickly.

VIII. DISCUSSION

A. Integration of the Emerging Model with the Literature

The components and indicators of problem-solving skills were evaluated by confirmatory factor analysis (CFA). It was related to the result of synthesization of problem-solving skills by literature review [20-25]. Based on theoretical studies and research articles related to problem-solving skills and robot programming, the researchers were able to summarize the five components of problem-solving skills: 1) identifying the problem, 2) setting goals, 3) creating a solution, 4) acting on the solution and 5) checking the results. In addition, the researchers concluded that the robot programming procedure consists of five steps: 1) identifying the problem, 2) designing a solution, 3) coding the program, 4) testing the program, and 5) implementing the program.

B. Reflection on Methodology and Limitations

Because problem solving skills are explicitly a latent variable, the researchers had to transform latent variables into observable variables by using the behavioral statement in the problem-solving process. In addition, the researchers apply the statement with samples only found in science/research institutions. This point is because data for accurate CFA processing in this research is required only from those expected to have the required skills in robot programming.

C. Suggestions for Future Research

1) This research uses the confirmatory factor analysis (CFA) technique to study the structural validity of the components and indicators. In the next step, creating a standardized skill measurement instrument means that information can be tested for content validity and the accuracy of the questions checked before applying them to each context.

2) In the case of constructivism learning theory, which is popular in education systems dealing with such aspects as problem-based learning (PBL), project-based learning (PjBL) or challenge-based learning (CBL) [58], all the focus is on developing higher order thinking skills. Therefore, the researcher suggests that the five components and fifteen

indicators identified in this study could be used in the design of learning and evaluation processes in modern education.

ACKNOWLEDGMENT

This research article was supported form "Petchra Pra Jom Klao Ph.D" Research Scholarship from King Mongkut's University of Technology Thonburi.

REFERENCES

- [1] OECD. (2020). Learning Compass 2030 - OECD Future of Education and Skills 2030. Oecd.org. Retrieved 5 January 2020, from <https://www.oecd.org/education/2030-project/teaching-and-learning/learning/learning-compass-2030/>.
- [2] OECD. (2020). Future shocks and shifts: challenges for the global workforce and skills development. Retrieved 5 January 2020, from <https://www.oecd.org/education/2030-project/about/documents/Future-Shocks-and-Shifts-Challenges-for-the-Global-Workforce-and-Skills-Development.pdf>.
- [3] Partner for 21st Century Skills, "Framework for 21st Century Learning. (2009). Framework for 21st Century Learning. Teacherrambo.com. Retrieved 5 January 2020, from https://www.teacherrambo.com/file.php/1/21st_century_skills.pdf.
- [4] Economic Forum Executive. (2018). The Future of Jobs Report 2018. World Economic Forum. Retrieved 5 January 2020, from <https://www.weforum.org/reports/the-future-of-jobs-report-2018>.
- [5] Canter, A. (2004). A Problem-Solving Model for Improving Student Achievement. Casponline.org. Retrieved 7 January 2020, from <https://www.casponline.org/pdfs/pdfs/rti0004.pdf>.
- [6] Bloom, B. (1984). Taxonomy of educational objectives. Longman.
- [7] TENOPYR, M., Guilford, J., & HOEPFNER, R. (1966). A factor analysis of symbolic memory abilities: studies of aptitudes of high-level personnel. By M. L. Tenopyr, J. P. Guilford and R. Hoepfner. University of Southern California, Psychological Laboratory.
- [8] Jonassen, D. (2011). Learning to solve problems. Routledge.
- [9] Institute for the Promotion of teaching Science and Technology. (2018). Summary of curriculum and indicators Information Technology and Communication Curriculum 2008 and Technology (Computational Science) Curriculum Improvement 2018. Retrieved 9 January 2020, from <http://oho.ipst.ac.th/download/mediaBook/cs-ict.pdf>.
- [10] National Science and Technology Development Agency. (2020). Space Flying Robot Programming Challenge 2020 (SRPC2020). NSTDA. Retrieved 9 January 2020, from <https://www.nstda.or.th/jaxa-thailand/krpc2020>.
- [11] Alimisis, D. (2013). Educational robotics: Open questions and new challenges. Themes In Science & Technology Education, 6(1), 63-71. Retrieved 10 January 2020, from https://www.researchgate.net/publication/284043695_Educational_robotics_Open_questions_and_new_challenges.
- [12] Kanbul, S., & Uzunboylu, H. (2017). Importance of Coding Education and Robotic Applications For Achieving 21st-Century Skills in North Cyprus. International Journal Of Emerging Technologies In Learning (Ijet), 12(01), 130. <https://doi.org/10.3991/ijet.v12i01.6097>.
- [13] Demertzi, E., Voukelatos, N., Papagerasimou, Y., & Drigas, A. (2018). Online Learning Facilities to Support Coding and Robotics Courses for Youth. International Journal Of Engineering Pedagogy (Ijep), 8(3), 69. <https://doi.org/10.3991/ijep.v8i3.8044>.
- [14] Tuomi, P., Multisilta, J., Saarikoski, P., & Suominen, J. (2017). Coding skills as a success factor for a society. Education And Information Technologies, 23(1), 419-434. <https://doi.org/10.1007/s10639-017-9611-4>.
- [15] Grover, S., & Basu, S. (2017). Measuring Student Learning in Introductory Block-Based Programming. Proceedings Of The 2017 ACM SIGCSE Technical Symposium On Computer Science Education. <https://doi.org/10.1145/3017680.3017723>.
- [16] Romero, M., Lepage, A., & Lille, B. (2017). Computational thinking development through creative programming in higher education.

- International Journal Of Educational Technology In Higher Education, 14(1). <https://doi.org/10.1186/s41239-017-0080-z>.
- [17] Silapachote, P., & Srisuphab, A. (2017). Engineering Courses on Computational Thinking Through Solving Problems in Artificial Intelligence. *International Journal Of Engineering Pedagogy (Ijep)*, 7(3), 34. <https://doi.org/10.3991/ijep.v7i3.6951>.
- [18] Topalli, D., & Cagiltay, N. (2018). Improving programming skills in engineering education through problem-based game projects with Scratch. *Computers & Education*, 120, 64-74. <https://doi.org/10.1016/j.compedu.2018.01.011>.
- [19] Lertyosbordin, C., Maneewan, S., Yampinij, S., & Thamwipat, K. (2019). Scoring Rubric of Problem-Solving on Computing Science Learning. *International Education Studies*, 12(8), 26. <https://doi.org/10.5539/ies.v12n8p26>.
- [20] Bransford, J., & Stein, B. (1993). THE IDEAL PROBLEM SOLVER A Guide for Improving Thinking, Learning, and Creativity. Tntech.edu. Retrieved 13 February 2020, from https://www.tntech.edu/cat/pdf/useful_links/idealproblemsolver.pdf.
- [21] Foshay, R., & Kirkley, J. (2003). Principles for Teaching Problem Solving. Vcell.ndsu.nodak.edu. Retrieved 13 February 2020, from http://vcell.ndsu.nodak.edu/~ganesh/seminar/2003_Foshay_PLATO%20Learning%20Inc._Tech%20Paper%20%234_Principles%20for%20Teaching%20Problem-Solving.pdf.
- [22] Great schools partnership. (2016). Performance Indicators for Problem Solving. Great schools partnership. Retrieved 14 February 2020, from <https://www.greatschoolspartnership.org/wp-content/uploads/2017/01/PDFTaskModelforProblemSolvingNov22-2016.pdf>.
- [23] Nurkaeti, N. (2021). Polya's Strategy: An Analysis of Mathematical Problem Solving Difficulty in 5th Grade Elementary School. Retrieved 2 May 2021, from <http://doi.org/10.17509/eh.v10i2.10868>.
- [24] Astuti, F., Suranto, S., & Masykuri, M. (2019). Augmented Reality for teaching science: Students' problem solving skill, motivation, and learning outcomes. *Jurnal Pendidikan Biologi Indonesia*, 5(2). <https://doi.org/10.22219/jpbi.v5i2.8455>.
- [25] Franestian, I., Suyanta, & Wiyono, A. (2020). Analysis problem solving skills of student in Junior High School. *Journal Of Physics: Conference Series*, 1440, 012089. <https://doi.org/10.1088/1742-6596/1440/1/012089>.
- [26] Fernández, E. (2015). Learning ROS for robotics programming. Packt Publishing.
- [27] Owen-Hill, A. (2018). Python vs C++ vs C# vs MATLAB: Which Robot Language is Best. robodk. Retrieved 10 March 2020, from <https://robodk.com/blog/robot-programming-language/>.
- [28] Plant Automation-technology. (2020). Different Types of Robot Programming Languages," Plant Automation-technology. Plantautomation-technology.com. Retrieved 12 March 2020, from <https://www.plantautomation-technology.com/articles/different-types-of-robot-programming-languages>.
- [29] The University of Sheffield. (2020). Programming skills for robotics. FutureLearn. Retrieved 13 March 2020, from <https://www.futurelearn.com/courses/robotic-future/0/steps/29368>.
- [30] Sharma, M. (2017). A Study of SDLC to Develop Well Engineered Software. *International Journal Of Advanced Research In Computer Science*, 8(3), 502-523. <https://doi.org/https://doi.org/10.26483/ijarcs.v8i3.3045>.
- [31] Valenzuela, J. (2018). Computer programming in 4 steps | ISTE. Iste.org. Retrieved 10 June 2020, from <https://www.iste.org/explore/Computer-Science/Computer-programming-in-4-steps>.
- [32] Wikibooks. (2021). The Computer Revolution/Programming/Five Steps of Programming - Wikibooks, open books for an open world. En.wikibooks.org. Retrieved 11 March 2020, from https://en.wikibooks.org/wiki/The_Computer_Revolution/Programming/Five_Steps_of_Programming.
- [33] Amjo. (2018). Six steps in the programming process | Dotnetlanguages. Dotnetlanguages.net. Retrieved 10 March 2020, from <https://www.dotnetlanguages.net/six-steps-in-the-programming-process/>.
- [34] Person, M. (2020). Programming Development Process: 4 Steps to Better Programming STEP 1: Define and Analyze the Problem. Academia.edu. Retrieved 13 May 2020, from https://www.academia.edu/11050507/Programming_Development_Process_4_Steps_to_Better_Programming_STEP_1_Define_and_Analyze_the_Problem.
- [35] Department of Computer Science and Statistics. Computer Programming. The University of Rhode Island. Retrieved 10 May 2020, from <https://homepage.cs.uri.edu/faculty/wolfe/book/Readings/Reading13.htm>.
- [36] School of Computer Science. The Programming Process. University of Birmingham. Retrieved 10 March 2020, from <https://www.cs.bham.ac.uk/~rxb/java/intro/2programming.html>.
- [37] Yamane, T. (1974). *Statistics ; an introductory analysis*. Harper & Row.
- [38] Wolf, E., Harrington, K., Clark, S., & Miller, M. (2013). Sample Size Requirements for Structural Equation Models. *Educational And Psychological Measurement*, 73(6), 913-934. <https://doi.org/10.1177/0013164413495237>.
- [39] Turner, R., & Carlson, L. (2003). Indexes of Item-Objective Congruence for Multidimensional Items. *International Journal Of Testing*, 3(2), 163-171. https://doi.org/http://doi.org/10.1207/S15327574IJT0302_5
- [40] Hair, J. (2010). *Multivariate data analysis*. Pearson Education.
- [41] Hooper, D., Coughlan, J., & Mullen, M. (2008). Structural Equation Modeling: Guidelines for Determining Model Fit. *Electronic Journal On Business Research Methods*, 6(1), 53-60. Retrieved 2 May 2021, from https://www.researchgate.net/publication/254742561_Structural_Equation_Modeling_Guidelines_for_Determining_Model_Fit.
- [42] Statistics Solutions. (2020). Factor Analysis - Statistics Solutions. Statistics Solutions. Retrieved 15 June 2020, from <https://www.statisticssolutions.com/factor-analysis-sem-factor-analysis>.
- [43] APA. (2020). APA Dictionary of Psychology. APA Dictionary. Retrieved 5 June 2020, from <https://dictionary.apa.org/problem-solving-approach>.
- [44] Chandrasekaran, B. (1990). Design Problem Solving: A Task Analysis. *AI Magazine*, 11(4), 59-71. <https://doi.org/https://doi.org/10.1609/aimag.v11i4.857>.
- [45] McFadden, C., McFadden, C., Wendorf, M., Engineering, I., & Bergan, B. (2020). How to Think like a Programmer When Problem Solving. Interestingengineering.com. Retrieved 13 May 2020, from <https://interestingengineering.com/how-to-think-like-a-programmer-when-problem-solving>.
- [46] Taheri, S., Sasaki, M., & Ngetha, H. (2015). Evaluating the effectiveness of problem solving techniques and tools in programming. 2015 Science And Information Conference (SAI). <https://doi.org/10.1109/sai.2015.7237253>.
- [47] Barlow-Jones, G., & van der Westhuizen, D. (2017). Problem Solving as a Predictor of Programming Performance. *Communications In Computer And Information Science*, 209-216. https://doi.org/10.1007/978-3-319-69670-6_14.
- [48] Kember, J. (2018). Problem Identification - FastBridge. FastBridge. Retrieved 11 June 2020, from <https://www.fastbridge.org/2018/01/problem-identification/>.
- [49] Staniewicz, S. (2013). Problem Identification in Engineering Design | Electrical and Computer Engineering Design Handbook. Sites.tufts.edu. Retrieved 15 June 2020, from <https://sites.tufts.edu/eenior/designhandbook/2013/problem-identification-in-engineering-design/>.
- [50] Wharton Executive Education. (2015). How to Identify the Real Problem - Nano Tools for Leaders. Wharton Executive Education. Retrieved 14 June 2020, from <https://executiveeducation.wharton.upenn.edu/thought-leadership/wharton-at-work/2015/06/identify-the-real-problem/>.
- [51] Markman, A. (2017). How You Define the Problem Determines Whether You Solve It. *Harvard Business Review*. Retrieved 18 June 2020, from <https://hbr.org/2017/06/how-you-define-the-problem-determines-whether-you-solve-it>.
- [52] Crews, T., & Ziegler, U. The flowchart interpreter for introductory programming courses. FIE '98. 28Th Annual Frontiers In Education Conference. Moving From 'Teacher-Centered' To 'Learner-Centered'

- Education. Conference Proceedings (Cat. No.98CH36214). <https://doi.org/10.1109/fie.1998.736854>.
- [53] Bryant, D. (2021). Critique, Explore, Compare, and Adapt (CECA): A New Model for Command Decision Making. DTIC. Retrieved 18 June 2020, from <https://apps.dtic.mil/sti/citations/ADA605875>.
- [54] Whipp, D., Tenkanen, H., & Heikinheimo, V. (2020). Good coding practices - Selecting “good” variable names. [Geo-python.github](https://geo-python.github.io/site/notebooks/L1/gcp-1-variable-naming.html). Retrieved 10 July 2020, from <https://geo-python.github.io/site/notebooks/L1/gcp-1-variable-naming.html>.
- [55] Bilotta, E., & Pantano, P. (2021). Some problems of Programming in Robotics (pp. 209-220). Cozenza, Italy. Retrieved 2 May 2021, from <https://www.ppig.org/files/2000-PPIG-12th-bilotta.pdf>.
- [56] Young, J., Sharlin, E., & Igarashi, T. (2013). Teaching Robots Style: Designing and Evaluating Style-by-Demonstration for Interactive Robotic Locomotion. *Human-Computer Interaction*, 28(5), 379-416. <https://doi.org/10.1080/07370024.2012.697046>.
- [57] Ficanha, E., Ribeiro, G., Dallali, H., & Rastgaar, M. (2016). Design and Preliminary Evaluation of a Two DOFs Cable-Driven Ankle-Foot Prosthesis with Active Dorsiflexion-Plantarflexion and Inversion-Eversion. *Frontiers In Bioengineering And Biotechnology*, 4. <https://doi.org/10.3389/fbioe.2016.00036>.
- [58] Lynce, M. (2017). What is the Difference Between Problem, Project, and Challenge Based Learning? - The Edvocate. The Edvocate. Retrieved 17 July 2020, from <https://www.theedadvocate.org/difference-problem-project-challenge-based-learning/>.