Chen, W. et al. (Eds.) (2017). Proceedings of the 25th International Conference on Computers in Education.
New Zealand: Asia-Pacific Society for Computers in Education

Computational Thinking Development through Programmable Robotics Activities in STEM Education in Primary Schools

Siu-Cheung KONG * & Chan-Chio LAO
Centre for Learning, Teaching and Technology,
The Education University of Hong Kong, Hong Kong
*sckong@eduhk.hk; acclao@eduhk.hk

Abstract: Programmable robotics activities in science, technology, engineering and mathematics (STEM) education have been postulated to have positive impacts on Computational Thinking (CT) development. This study aims to discuss how these activities in primary school STEM education should be designed to nurture students with CT abilities. According to Sullivan and Heffernan (2016), CT learning progression with robotics consists of four stages, namely sequencing, causal inference, conditional reasoning, and systems thinking. Three examples about auto-piloting a robotic car to 1) run a square, 2) run along the white track, and 3) slow down and stop when it detects an obstacle are designed to illustrate the learning progression. The first example provides opportunities for students to develop their abilities of sequencing and causal inference. The second example demonstrates how conditional reasoning can be developed. The third example shows how systems thinking can be established. Based on this learning progression for CT development, an outline of STEM education with programmable robotics activities in formal and non-formal learning in primary schools is proposed. The key is that problem-solving should be the core of these STEM activities. Students' knowledge in STEM related subjects in primary schools like science, mathematics and programming should be applied and in turn being consolidated.

Keywords: Computational thinking, primary schools, programming, robotics, STEM education

1. Introduction

Science, technology, engineering and mathematics (STEM) education has become increasingly popular in recent years. The reasons of such popularity are that STEM relates to a wide range of professions practiced in the physical world (Weintrop et al., 2016) and it can develop student's realistic view for pursuing careers in the future (Committee on Prospering in the Global Economy of the 21st Century, 2007). Computational Thinking (CT), which is defined as "the thought processes involved in formulating problems and their solutions so that the solutions are represented in a form that can be effectively carried out by an information-processing agent" (Cuny, Snyder, and Wing, 2010, p.1), is closely related to STEM education. It is regarded as a core of all STEM activities and permeates all aspects of STEM (Basu et al., 2016). In addition, it acts as a consolidation of STEM experiences by deepening the understanding of STEM content areas for students (Sengupta, Kinnebrew, Basu, Biswas, and Clark, 2013). For example, STEM concepts like friction and environmental conservation are more understandable through engaging in programming and CT modelling activities (Hambrusch, Hoffmann, Korb, Haugan, and Hosking, 2009).

Although there are discussions about STEM and CT education (e.g., Sengupta, 2011; Sullivan and Heffernan, 2016; Weintrop et al., 2016), there have been few attempts to discuss how to conduct STEM education for CT development in primary school education. In particular, most of the previous studies have not emphasized how students might benefit from working with programmable robotics activities in primary school STEM education. Robotics focuses on physical object constructions, which could enhance students' incentives in learning STEM-related concepts, confidence, interest and academic achievements (Park, 2015). Following the CT learning progression with robotics suggested by Sullivan and Heffernan (2016), this study illustrates some examples of programmable robotics activities that primary schools could be designed. Based on this learning progression, an outline of STEM education using programmable robotics for CT development in formal or non-formal learning in primary schools is proposed for nurturing the next generation to become creative problem-solvers.

2. Computational Thinking in STEM Education

Sullivan and Heffernan (2016) discussed how robotic construction kits (RCKs) operate as computational manipulatives in P-12 STEM education after an extensive review of literature. CT could be developed in STEM education through robotics activities, which engage students with hands-on experience by building a model, interact with the robot through coding. Based on the data collected from environmental sensors such as ultrasonic sensor, line follower sensor, or input modules like switches and buttons, the robots will function automatically according to the instructions set in the computational environment. They would react via output modules such as wheels, motor and buzzers.

Students might use RCKs such as LEGO Mindstorms and mBot for conducting robotics activities. In the process of using RCKs for solving STEM related problems, students can work with these computational manipulatives. Therefore, Sullivan and Heffernan (2016) proposed a four-stage learning progression in educational robotics activities for CT development. It is expected that students would acquire and demonstrate ability of the previous stage before moving to the next. Students could go through a progression from fundamental sequencing ability to reasoning abilities, including causal inference and conditional reasoning. The highest level of this progression is that students' systems thinking will be improved. Table 1 shows the detailed explanations of each stage. By engaging in the robotics activities, students could also enhance their problem-solving skills by elementary approach like trial and error to more complex modeling method (Sullivan and Heffernan, 2016).

Table 1: Robotics learning progression for CT development proposed by Sullivan and Heffernan (2016).

Stage	CT Progression	Explanations
1	Sequencing	- Ability to put items into specific order
2	Causal inference	- Comparison between the expected movement and immediate feedback from the programmed object - Hypothesis on why expected movement was not observed
3	Conditional reasoning	- Abstraction of a rule for the behavior - Use of environmental sensors to work with robotic device such as sensory-reason-action loop
4	Systems thinking	- Understanding of interacting functions of related parts of robotic device - Interaction between sensors, microcomputer and actuators such as motor and bulbs

3. Examples Illustrating CT Progression Development through Programmable Robotics in STEM Education

The programmable robotics activities illustrated in this section were designed as group learning activities for Primary 5 students. These learning activities were successfully piloted in two primary schools in Hong Kong, in which six student groups were divided in each participating Primary 5 class with around 30 students for experiencing the learning progression of sequencing, causal inference, conditional reasoning, and systems thinking in the process of CT development.

3.1 Sequencing

Sequencing means the ability to arrange items into specific order to accomplish tasks with specific intention (Sullivan and Heffernan, 2016). Primary school students could develop their sequencing ability by arranging blocks into specific orders in programming environment. For example, students can be asked to develop a program to auto-pilot a robotic car to run in a square as shown in Figure 1(a).

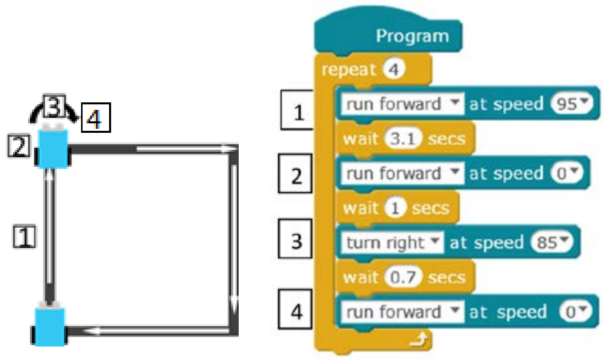

Figure 1(a). An example that illustrates the sequencing ability of students by auto-piloting a robotic car to run a square with codes in a generalizable pattern.

Figure 1(b). An example that illustrates the sequencing ability of students by auto-piloting a robotic car to run a square with codes not in a generalizable pattern.

To accomplish this task, students have to remote control the car to run in a square. The following four steps in sequential order are needed: Step 1, run forward for a certain period of time (remark: this is the length of the side of a square); Step 2, stop running for a short period of time (remark: the car takes time to stop properly); Step 3, turn right for a period of time with a certain speed so that the car can turn right for 90 degrees from the original direction; and Step 4, stop running (remark: the car has to stop properly). The robotic car will run in a square and return to its initial position if the above sequences of steps are repeated for four times. However, when students start producing codes to run in a square, lines of codes as shown in Figure 1(b) are commonly found. The car might run forward with some steps and then more steps forward to complete running along one side of the square. The turning to right for 90 degrees procedures might be decomposed into turning to the right for 30 degrees and then 60 degrees. There is a need for teachers to guide students to learn from the stage with codes not ready to generalize a pattern to the stage of producing codes in running a square with pattern as shown in Figure 1(a).

3.2 Causal Inference

When students arrange the steps in order, they will run and test the programs at the same time. Causal inference will take place in this process. After comparing the actual movement with the intended movement of the robotic car, students would hypothesize the cause of such discrepancies. Using the same example in section 3.1 (piloting a robotic car to run in a square), causal inference is demonstrated in Figure 2.

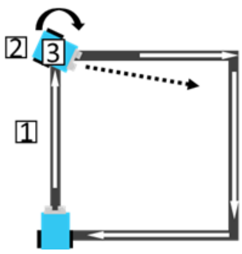

Figure 2. An example illustrating the ability of students in causal inference by auto-piloting a robotic car to right turn 90 degrees.

Students might discover that the robotic car fails to turn 90 degrees as expected (Step 2 in Figure 2). Such discrepancy might make the robotic car fail to run in a square (Step 3 in Figure 2). When the actual movement is not in line with the expected outcome, students may have different interpretations of the discrepancies. Students who demonstrate the ability of causal inference will be able to identify the fault sources and rectify them. In this case, the speed value of right turn and the duration of this process in the program have to be adjusted in order to find out the accurate parameters for turning 90 degrees in this case. After iterative inference and modification of the program, students might find out a possible solution. This experience can help students to develop the ability of causal inference.

3.3 Conditional Reasoning

Conditional reasoning refers to the ability to abstract rules to confine behavior in an environment. It involves logical reasoning in a programmable environment using data collected from sensors for defining proper operation of devices in the environment. Robots will follow the sensory-reason-action-loop computation. In other words, they would react to computation results based on the sensor input and take actions accordingly. In this example, students are asked to develop a program to make the robotic car move forward along the white track (see Figure 3(a)). In this task, results of the line follower sensors are imported to port 2 of the robotic car.

Figure 3(b) shows the value returned by the two-line follower sensors. Based on the sensor inputs, that is the values returned by the line follower sensors, students need to assign appropriate actions to the inputs. If the robotic car is not on the track and with the returned value of 0, the robotic car is programmed to stop. If the robotic car is touching the black edge on the left and with the returned value of 1, the robotic car is programmed to turn right. If the robotic car is touching the black edge on the right and with the returned value of 2, the robotic car is programmed to turn left. If the robotic car is exactly on the white track and with the returned value of 3, the robotic car is programmed to move forward. In this example, a sensory-reason-action loop is constructed using multiple if-then conditionals.

	Left sensor color	Right sensor color	Returned value
	Black	Black	0
	Black	White	1
	White	Black	2
	White	White	3

Figure 3(a). An example of preparing a coding table for conditional reasoning in piloting a robotic car to run along the white track.

Figure 3(b). An example of a piece of codes illustrating implementation of conditional reasoning in piloting a robotic car to run along the white track.

3.4 Systems Thinking

Systems thinking refers to “judgement and decision making, systems analysis, evaluation and abstract reasoning” about the ways that different elements interact in a system (Sardone, 2017, p. 34). It offers a bird eye’s view of looking at connections in the world dynamically (Breil, Ritchie, and Greer, 2017). In the context of robotics, systems thinking refer to the understanding of functions, hierarchy, relationship, and interaction among sensors, microcomputers, actuators such as motor and bulbs, and other related components in the systems. The example in section 3.3 is further discussed in this section to illustrate the ability of systems thinking. Figure 4 demonstrates the adding of a program requirement to the task. The addition tasks of the robotic car are to slow down when an obstacle is detected afar and stop when an obstacle is nearby along the white track. Ultrasonic sensor is imported to port 3 of the robotic car to attain this goal.

In this example, an additional set of codes is added to the system. When the returned value of the line follower sensors is equal to 3 (i.e. the robotic car is moving along the white track), then a further checking of the two conditions is needed. First, if the robotic car detects an obstacle in less than 50 units from itself, it will slow down (see codes in section A of Figure 4). Second, if the robotic car further detects an obstacle in less than 20 units, it will stop to avoid crashing with the obstacle (see codes in section B of Figure 4). In this example, the task requires students to understand interactions among the line follower sensors, ultrasonic sensor, wheels and motor (speed) of the robotic car, and the microcomputer (Arduino in this example) of the system in order to program the robotic car to function appropriately. This example illustrates that students possess an understanding of the hierarchy and the relationship of different robotic elements if they are able to construct a program to accomplish the tasks and these are the core elements to indicate students' systems thinking ability.

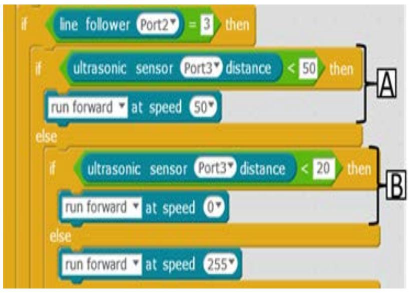

Figure 4. An example illustrating the ability of systems thinking of students in auto-piloting a robotic car to move along the white track and additionally need to slow down and stop when an obstacle is detected.

4. An Outline of STEM Education Using Programmable Robotics Activities for CT Development in Formal and Non-Formal Learning in Primary Schools

In a bid to address STEM education for CT development in primary schools, this study proposes an outline of STEM education using programmable robotics activities. There are two possible opportunities for primary school students to develop CT in STEM education. One way is to incorporate STEM activities into formal curriculum subject learning such as science, mathematics, and programming in primary school curriculum. Another way is to promote STEM activities in non-formal learning such as co-curricular activities. Following Sullivan and Heffernan's framework (2016) of robotics learning progression, Figure 5 shows the proposed outline.

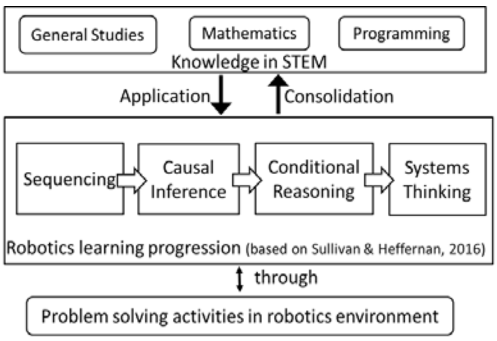

Figure 5. An outline of STEM education using programmable robotics activities for CT development in formal and non-formal learning in primary schools.

The key is that problem-solving should be the core of these STEM activities. Students' knowledge in STEM related subjects in primary schools like science, mathematics and programming should be applied and in return being consolidated. Through engaging students' in problem-solving activities such as auto-piloting a robotic car in formal or non-formal programmable robotics doings, students' knowledge in science such as friction, mathematics like relationship between speed and distance, and programming concepts of conditionals and loops need to be applied. These concepts would probably in return being consolidated through these STEM activities. Students are expected to follow the learning progression of sequencing, causal inference, conditional reasoning, and systems thinking in the process of CT development. Experiences tell that students rarely program correctly at the first instance. In general, they need rounds of attempts such as adjusting the speed and direction of turning of the robotic car before they can program the car to run according to the instructions. Thus, students might demonstrate their abilities of sequencing, causal inference and conditional reasoning iteratively and interactively in the learning progression. Students might have the idea of systems thinking when they have experience of accomplishing the tasks of running along the white track and avoiding clashes with obstacles along the way. Students then can be encouraged to think creatively on how to improve further the functions of the robotic cars if more sensors can be introduced. This is a way to nurture them to become creative problem-solvers through this CT development process.

The goal of STEM education through programmable robotics activities is CT development. Examples in this study illustrate how STEM education in the context of solving robotics related problems in primary schools can be linked with the existing primary school curriculum for supporting the development of CT among primary school students. The robotics activities can provide opportunities for students to enhance confidence in applying STEM-related subject knowledge into solving real life problems in robotics context. Further work is to implement these activities in school formal or non-formal learning and conduct evaluation study on CT development.

Acknowledgements

The authors would like to acknowledge the support of the coolthink@JC project funded by the Hong Kong Jockey Club Charities Trust.

References

Basu, S., Biswas, G., Sengupta, P., Dickes, A., Kinnebrew, J. S., & Clark, D. (2016). Identifying middle school students' challenges in computational thinking-based science learning. *Research and Practice in Technology Enhanced Learning, 11*(1), 13.

Breil, B., Ritchie, T., & Greer, H. (2017). Learning and applying systems thinking when studying a local creek system. *Science Scope, 40*(6), 33-40.

Committee on Prospering in the Global Economy of the 21st Century: An Agenda for American Science and Technology. (2007). *Rising above the gathering storm: Energizing and employing America for a brighter economic future*. Washington, DC: National Academies Press.

Cuny, J., Snyder, L., & Wing, J. M. (2010). Demystifying computational thinking for non-computer scientists. Unpublished manuscript in progress, referenced in http://www.cs.cmu.edu/~CompThink/resources/TheLinkWing.pdf.

Hambrusch, S., Hoffmann, C., Korb, J. T., Haugan, M., & Hosking, A. L. (2009, March). A multidisciplinary approach towards computational thinking for science majors. In *Proceedings of the 40th ACM Technical symposium on Computer Science Education* (pp. 183-187). New York, NY: ACM.

Park, J. (2015). Effect of robotics-enhanced inquiry-based learning in elementary science education. *Journal of Computers in Mathematics and Science Teaching, 34*(1), 71-95.

Sardone, N. B. (2017). Building bots to develop systems thinking. *Science Scope, 40*(5), 32-41.

Sengupta, P. (2011, June-July). Design principles for a visual programming language to integrate agent-based modeling in K-12 Science. In *Proceedings of the Eighth International Conference of Complex Systems* (pp. 1636-1637). Quincy, MA: New England Complex Systems Institute.

Sengupta, P., Kinnebrew, J. S., Basu, S., Biswas, G., & Clark, D. (2013). Integrating computational thinking with K-12 science education using agent-based computation: A theoretical framework. *Education and Information Technologies, 18*(2), 351-380.

Sullivan, F. R., & Heffernan, J. (2016). Robotic construction kits as computational manipulatives for learning in the STEM disciplines. *Journal of Research on Technology in Education, 48*(2), 105-128.

Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining computational thinking for mathematics and science classrooms. *Journal of Science Education and Technology, 25*(1), 127-147.