10

EXAMINING THE MULTIDIMENSIONAL LEARNING AFFORDANCES OF ROBOTICS FOR COMPUTATIONAL THINKING AND SCIENCE INQUIRY

Florence R. Sullivan

ROBOTICS PROBLEM SPACES

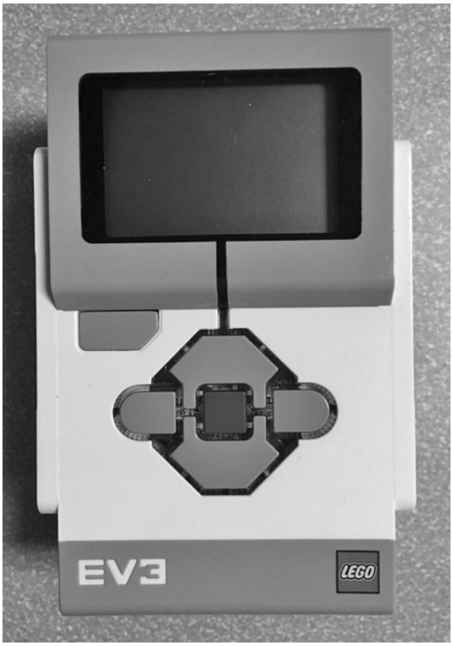
Computational thinking (CT) is an integral aspect of learning and work in the science, engineering, technology, and mathematics (STEM) fields (Lee et al. 2020). Indeed, the Next Generation Science Standards (NGSS) (2013) have defined mathematics and CT as one of the eight core disciplinary practices of science activity. Robotics is a robust learning environment that supports the development of CT and science literacy (Sullivan 2008; Sullivan and Heffernan 2016). Foundational to robotics learning is integrated interaction in the three problem spaces typical of all robotics learning environments, including the device itself, the screen-based programming environment, and the actual physical environment in which students are testing their robotic device. This chapter begins with a description of each of the problem spaces, individually, and proceeds with examples of student learning drawn from fifteen years of research on the topic. Specifically, I discuss student engagement in both science literacy practices (e.g., systems thinking, inferential reasoning) and CT practices (e.g., abstraction, creative problem solving, and algorithmic thinking) as both are supported by engagement in robotics learning. The chapter concludes with thoughts for future research directions. These observations derive from both cognitive and sociocultural viewpoints, with early work grounded in task analysis and mental representations (Roth 2001), and later work grounded in a socio-cultural framework (Vygotsky 1978).

THE FIRST PROBLEM SPACE: THE ROBOTIC DEVICE

We have focused primarily on using the LEGO Mindstorms robotics kit with students. Therefore, in this chapter, I describe this device as the first problem space. However, any robotic device that includes the same elements as the LEGO kits will support student learning in the same way. A micro-computer, called a brick, is at the heart of the LEGO Mindstorms kit; the brick was developed at the MIT Media Lab in the mid-1990s (Resnick et al. 1996). This brick, which is in its third iteration, is currently called the EV3. The EV3 is a device that can fit into the palm of an adult's hand (see figure 10.1). The brick has four ports in which output devices, such as servo motors, can be plugged in with connecting wires, and another four ports in which input devices, such as digital sensors, can be connected. There are three motors that come with the kit, two large motors and one small motor. The larger motors are typically used when children are building a vehicular robot. Once the vehicular robot is constructed, the motors are attached to wheels, and as the motor spins, so do the wheels. The third, smaller motor can be used to operate a robotic arm that may be affixed to the vehicular robot. While building a robotic vehicle is a popular approach, many other types of machines can be built with the materials.

In addition to the brick and the motors, each robotic kit comes with several digital sensors, including a color sensor, a touch sensor, and an ultrasonic sensor. These sensors can be used in one of two ways (both of which are important for science inquiry and are discussed in greater detail later). The first mode is a data collection and display mode; the second is a wait-for mode that can trigger a specific event, once a threshold has been met or crossed. The kit also includes a number of LEGO pieces, called Technics, which fit together around the brick and the motors to create any number of structures or vehicles.

The design of the robotic device is dictated by the challenge that students are attempting to solve. As noted previously, often a robotic vehicle

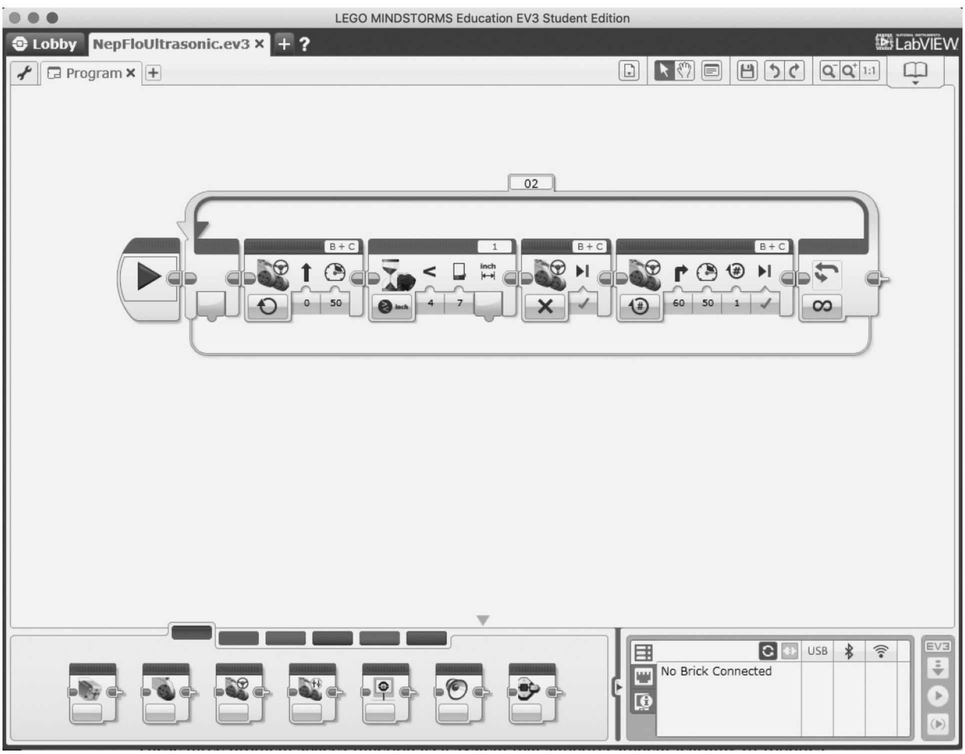


10.1 The LEGO Mindstorms EV3.

is constructed and sensors are then added to the vehicle to aid in navigation. For example, the ultrasonic sensor measures the distance between the sensor and objects in its path; using this sensor, a program can be written that will allow the robot to circumnavigate obstacles in the room. The design of the robot as a problem space revolves around accurate design, physical construction, and correct wiring of the motors and the sensors. While students may initially develop a robotic device that they think is adequate, through the process of working out a solution to the given challenge, students will often need to revise their design. So, while we may think of the design of the device as the first problem space, it is a problem space that is returned to throughout the duration of problem-solving activity.

THE SECOND PROBLEM SPACE: THE SCREEN-BASED PROGRAMMING ENVIRONMENT

At this point, several types of software can be used to program the LEGO EV3 robot: the actual software created by LEGO called LabVIEW for LEGO MINDSTORMS (LVLM); an extension that can be used in the 2-D animation and game programming environment, Scratch (Scratch, n.d.); EV3python; RobotC; and other programming environments (LEGO Engineering, n.d.). For the purposes of this chapter, I focus our discussion by drawing examples from LVLM. LVLM (see figure 10.2) is designed as a drag-and-drop, block-based programming environment. It provides action blocks for programming output devices (motors, sound, display, and/or the brick light), flow control blocks for programming wait for loops and sensor triggered events, sensor blocks for additional programming of sensors



10.2 The LabVIEW for LEGO Mindstorms (LVLM) programming environment.

including data collection, data operations blocks for working with data that have been collected, advanced blocks (including message blocks, and Bluetooth-enabled operations), and finally a “My Blocks” section, where users can create their own blocks.

In addition to the programming blocks, the software includes a utility in the bottom right-hand corner of the interface that, when the EV3 is connected to the laptop, allows the user to quickly verify which ports the motors are connected through, which port a sensor is connected to, and whether that sensor is actually reading environmental data. This, along with a context-sensitive help utility, which can be selected from a drop-down menu, allows students to learn about the programming environment and also verify that all parts of the robot are functional.

THE THIRD PROBLEM SPACE: THE PHYSICAL ENVIRONMENT

For the purposes of this chapter I describe a specific environment, developed by the FIRST LEGO League, which is an international, nonprofit organization that publishes a thematic robotics challenge and holds regional robotics events each year in which children participate. While this is a specific physical environment, the reader should bear in mind that robotics environments can be created in any room, and/or one could do robotics outdoors. Indeed, any physical space could be a potential robotics environment. The FIRST LEGO League challenge map is four feet wide by eight feet wide, which can be laid on the floor or set on a table with similar dimensions. The challenge map comes with specific pieces that are placed in specific spots on the map. For the purposes of this chapter, I provide an image of one such challenge map created by the FIRST LEGO league (2011). This challenge map was used in 2011 and is known as the Food Factor Challenge (see figure 10.3). In this challenge, children were tasked with completing specific large-scale food production robotic tasks on the challenge board, while considering the environmental effects of such production (e.g., the long-term effects of over-fishing). The board consists of fifteen different challenges. All of the challenges include a description of a real-world problem that the challenge attempts to solve.

10.3 The Food Factor Challenge Board by FIRST LEGO League.

LEARNING IN THE MULTIDIMENSIONAL PROBLEM SPACE

From a Vygotskyan (1978) perspective, students learn in the robotics environment through interaction with the tools and dialogue with each other and the teacher. It is important to note that the learning outcomes described in the following pages are made possible through a pedagogical approach that affords open-ended, collaborative learning. It is children's free movement within the space that also contributes to their learning (Dewey 1938/1997). In other words, while children should be given a specific challenge to solve, within the activity itself, children should have freedom to explore various solutions and various approaches. It is through collaborative exploration that children are able to engage in practices that support their learning. In our research, we have found support for student learning and growth in the following areas: systems learning, science literacy, inferential reasoning, abstraction, creative problem-solving (including the role of play), problem-solving strategy development, and computational concepts (Sullivan 2008, 2011; Sullivan and Keith 2018; Sullivan and Lin 2012; Sullivan, Söken, and Yildiz 2019). This learning and growth are supported by the design affordances of the multidimensional robotics environment. I address each aspect of learning with robotics in turn.

SYSTEMS LEARNING

A system is defined as a collection of parts or processes (Penner 2000). Hmelo-Silver, Holton, and Kolodner (2000) define a complex system as one in which part of a system interacts with other systems; to understand a complex system, students must engage with the "causal interactions and functional relations" (p. 248) among systems. The three problem spaces that make up the robotics learning environment function as a complex system (Sullivan 2008). This is so because each problem space can be seen as a system in its own right. And, while the problem spaces are tightly coupled to create the learning environment, one must often master and troubleshoot errors in each system, as well as across the complex system, to solve challenges. For example, students often build a vehicular robot with the LEGO pieces and wheels when they are working with robotics. If the vehicle is constructed poorly, it will affect the performance of the entire system. Therefore, students would need to work on fixing the building error to continue with any challenge solution.

Meanwhile, the program may contain an error that prohibits it from executing when transferred to the robot. In this instance, the feedback students receive is simply no feedback: the robot will not execute the program, it will not move. Students then must return to the programming space to puzzle through the error. Importantly, students are learning about the robotic system through these debugging activities. In this way, it is easy to see how learning to think computationally (debugging a robotics problem) is connected to science inquiry (learning about systems). In our prior research, we found that students' understanding of systems improved after a long summer course in robotics. A total of twenty-six fifth-grade students, ages ten to twelve years, worked in a three-week, 105-hour robotics course. Results on a systems thinking test created by Cooper (2004) indicated that students' ability to think about systems improved significantly from before to after (Sullivan 2008).

SCIENCE LITERACY

Science literacy has been variously defined as the ability to engage in the activity of inquiry, including "making observations, posing questions, planning investigations, reviewing what is already known in light of experimental evidence, using tools to gather, analyze, and interpret data, proposing answers, explanations, and predictions; and communicating the results" (National Research Council [NRC] 1996, 23). Science literacy as defined by the Next Generation Science Standards (NGSS 2013) includes knowledge of disciplinary core ideas (specific to each area of science), science and engineering practices (including the practices identified previously by the NRC), and cross-cutting concepts (including concepts that apply to all domains of science). In robotics learning environments, students have the opportunity to engage in many of the practices defined by the NRC and the NGSS. In our prior research (Sullivan 2008), we identified some of the cross-cutting concepts students engage with, including cause and effect, systems and system models, and structure and function. For example, we found that the feedback loop created by the activity of writing and executing programs on the robotic device (problem spaces one and two) support student engagement with cause and effect, whereas building a robotic device to carry out specific tasks in a specific environment (problem spaces one and three) supports engagement with the concepts of structure and function. Finally, as noted earlier, students engage with and improve their understanding of the concept of systems as they work in the robotics learning environment (Sullivan 2007, 2008).

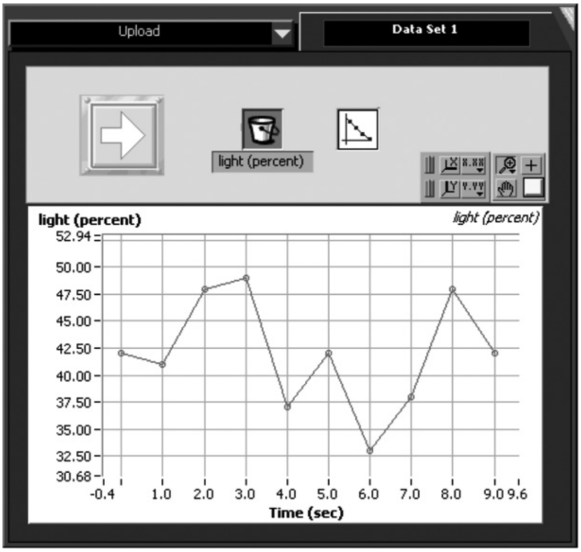
The NGSS (2013) refers to science and engineering practices as including observing, questioning, and planning, as well as designing, testing designs, analyzing results, and modifying the design accordingly. Importantly, these practices fall well within the CT construct as defined by other researchers (Barr and Stephenson 2011; International Society of Technology in Education and the Computer Science Teaching Association 2011). For example, planning is an aspect of problem-solving; designing is an aspect of programming activity; and testing designs, analyzing results, and revising designs constitute debugging activity.

In prior research, I identified a very regular set of activities that students engage in while working with robotics, which I have termed the troubleshooting cycle (TSC) (Sullivan 2011). The TSC consists of designing and building the robotic device, writing a program for the device, testing the program, diagnosing errors, debugging the program, and/or revising the design of the device, and retesting the program. This iterative practice encompasses action and interaction across the three problem spaces. The duration of a TSC is variable, it can last a few minutes, several minutes, or longer. However, the actual troubleshooting activity is very stable, it always consists of these six activities, and so it is an excellent unit of analysis for educational research; it can also serve to organize and support student learning and activity. For example, in one curricular implementation we studied, the teacher developed a note-taking worksheet that prompted students to record their trials, including what the students did, the problems they encountered, and their solutions to the problem. This worksheet is akin to a researcher's journal (Sullivan 2007). The troubleshooting cycle is a computational activity that is clearly an aspect of science and engineering practice as identified by the NGSS.

INFERENTIAL REASONING

As noted in the NGSS, "cause and effect" is a cross-cutting concept in science. In our research, we have found that interaction across the three problem spaces of robotics supports both hypothesis development, through debugging activity (Sullivan 2008), and inferential reasoning with data collected by sensors attached to the robotic device (Sullivan, Söken, and Yildiz 2019). Indeed, we have found that the sensors play an instrumental role in supporting student engagement in science and engineering practices in the robotics setting. The sensors are designed to monitor and/or collect data in the physical environment (the third problem space). The robotic device can be programmed to respond to a specific result when sensors are used to monitor the environment. The device can also be used as a means of collecting, storing, and then transmitting data to another device. In this way, the device, equipped with a programmed sensor can function as a scientific instrument for data collection.

We conducted a study in a sixth-grade science classroom, in which we followed a focal group of students as they worked to solve challenges that centered on heat and light energy topics (Sullivan, Söken, and Yildiz 2019). The students in the study were twelve years old; they were working with the second LEGO iteration of the brick (called the NXT) and a programming environment created at Tufts University called Robolab. Robolab is equipped with science investigation utilities, including a data graphing capability that allowed students to interpret the data numerically or through creating various graph-based visualizations of the data (see figure 10.4). The challenge the students were solving in this class was called Cave Explorer. This challenge asked students to explore three simulated cave environments to find out which one may be the most



10.4 Screenshot of the Robolab Datalogger.

comfortable to sleep in; the three simulated caves were actually three cardboard boxes, prepared with varying levels of light and heat inside. Students designed their robots with light, heat, and touch sensors and programmed them to navigate into the caves, collect data, and navigate back out. Table 10.1 presents a conversation among the students as one of them makes an inference from the data collected by the light sensor for one of the caves.

As can be seen in table 10.1, S makes an observation related to the differences in the numerical readings and then she makes an inference about where the data was collected. In line one, S has decided that the last three collected readings were collected outside of the cave, because of the numerical difference in the first three numbers as compared to the rest of the numbers in the data readout. Each of the "caves" was darker than the actual classroom itself. So, she infers that the light readings that were significantly higher in number were collected outside of the cave. Meanwhile, J interprets the last two readings as being outside the cave. In line seven, S notes that it is not just the last three but also the first light reading that was taken outside of the cave. In line 12, S begins to explain her reasoning to I (the third student in the group). While S is consistently interrupted by J, we

Table 10.1 School A student discussion—Making inferences from numerical data

Line	Speaker	Utterance	Researcher interpretation
1	S:	The, the last three [readings] are from outside.	Sara reads a numerical presentation of the collected light data and makes an inference based on it.
2	J:	What?	Javier asks Sara to repeat herself.
3	S:	The last three make, I think they're from outside because you know how when they came out there was two separate readings?	Sara repeats the comment and expands with some reasoning.
4	J:	No, the last two.	Javier interprets the data slightly differently.
5	S:	The last three.	Sara repeats claim.
6	J:	The last two.	Javier repeats claim.
7	S:	Three and then the first.	Sara continues to read the displayed data and interpret.
8	J:	Mister we got five hundred and two readings, why?	Javier asks the teacher a question about the printout.
9	S:	Yeah.	Sara affirms question.
10	T:	Oh, you got (?)	Teacher remark is partly unintelligible.
11	J:	You do it go and do it.	Javier instructs Sara to continue.
12	S:	Yeah, you know you're inside you're inside look, look he came out Ilana this . . .	Sara interprets the readings for Ilana.
13	J:	No don't (show it her) cause she's gonna say that's not gonna work.	Javier interferes with Sara's interpretation to Ilana.
14	S:	Look at this look at the light.	Sara continues interpreting.
15	J:	It's not gonna work.	Javier continues to interfere.
16	S:	These two are from outside.	Sara continues interpreting.
17	J:	It's not gonna work.	Javier continues to interfere.

(continued)

Table 10.1 (continued)

Line	Speaker	Utterance	Researcher interpretation
18	S:	And then . . .	Sara continues interpreting.
19	J:	It's not gonna work.	Javier continues to interfere.
20	I:	So, we got to do it all over again?	Ilana expresses confusion between Sara and Javier's comments.
21	J:	No.	Javier continues to interfere.
22	S:	And then these last these last three are from outside, and so feels right.	Sara continues interpreting and suggests the last cave "feels right."

can see that in lines 12, 14, 16, and 22, S points out to I how the amounts of reflected light are different and how that indicates where the readings were taken. In this example, it is possible to see that S is making inferences from the data. She is engaged in deductive reasoning from the data, and she is engaging in the cross-cutting concept of cause and effect—since the device is outside of the box, the light readings are higher. This is a powerful learning moment for these students that included both CT and science literacy elements. It is made possible by virtue of working in the multidimensional problem space of robotics; each of the problem spaces mattered in this interpretation, the designed device, the data read-out (part of problem space two), and the physical "cave" in which the robot collected data.

ABSTRACTION

In addition to supporting systems thinking and science literacy practices, the multidimensional problem space and iterative nature of robotics support the process of abstraction. Abstraction is an important computational concept. Abstraction refers to the stripping away of detail to reduce the complexity involved in a problem. The goal in abstraction is to identify the generalizable elements of a problem, which may be seen as foundational. It is when the foundational elements are clear that new representations of the problem can be developed, and these new representations can help lead to solutions. The three problem spaces of the robotics learning environment support abstraction in an *after the fact* mode. This is so because the physical robot and physical environment constitute 3-D representations of the problem, and the 2-D programming space offers an abstract representation of the 3-D movement of the robot. While working in the troubleshooting cycle, students move back and forth between the 3-D challenge environment and the 2-D programming environment. As they do so, they reason about the program they have written and the movement of the robotic device. In this way, the shift in attention, back and forth between the 2-D representation to the 3-D representation, supports students' model development and abstract thinking ability. Since the 2-D environment is provided to students, they do not have to create the abstraction (hence the after-the-fact mode). However, they do need to learn how to interpret the abstraction, and this work is supported by the 3-D aspects of the activity.

We have observed this behavior over and over again in our work. To demonstrate the phenomenon, we provide a vignette from a recent study (Sullivan and Keith 2018). Seventeen girls (ages eight to fourteen) participated in this case study. The case study focused on girls learning robotics in a one-day introduction to the FIRST LEGO league. Students worked collaboratively in groups of two or three to solve the challenges provided. Table 10.2 presents a short vignette featuring a conversation that one focal group of students had as they worked to solve a challenge. The conversation begins at the challenge board (lines 1 to 8), as the group observes the functioning of the robot, and continues as they move back to their worktable, where they were programming their robot.

As can be seen in table 10.2, the vignette begins with the students testing their robot. It does not work completely (lines 2–8), so they diagnose the problem, and then they move back to the 2-D representation and, as can be seen in line 17, L gesture and talk through what each icon programs the robot to do. While they are talking through the program, they are thinking back to what they just saw happen on the 3-D challenge board. In line 18, F pinpoints the block she believes should be programmed differently. It is this same activity that supports the students' ability to think more abstractly about the problem—each time the students execute the program, they must re-examine the icons used to program the robot to gain a better understanding of how to revise the program. This constant interplay between the 2-D and 3-D aspects of the activity provides students

Table 10.2 Abstraction dialogue

Line	Student	Utterance	Location	Researcher interpretation
1	L:	Okay, try that, I think that might have been what we have.	Challenge Board	Three students stand around the game board to test their executable program.
2	F:	Yeah, I think we just need to make *that* distance longer. What? Okay.	Challenge Board	Possible solution is forwarded by F. F is surprised by the robot's movement.
3	L:	Well . . .	Challenge Board	L makes an utterance while watching the robot.
4	F:	No.	Challenge Board	F articulates the failure of the program.
5	S:	It's crashing.	Challenge Board	S narrates the movement of the robot.
6	F:	Alright let's fix that.	Walking toward work table	F suggests group activity.
7	L:	Okay, what do we need to switch?	Challenge Board	L asks aloud what needs to be done.
8	S:	Okay, we need to make things that when it goes that way it's longer.	Challenge Board	S offers a potential solution.
9	F:	Yeah, we need one of the distances to be longer.	Walking toward worktable	F agrees with S's analysis.
10	S:	Haba	Worktable	S tries to sit in F's chair.
11	F:	S!	Worktable	F asks S to move (with tone implies S should quit fooling around).
12	L:	S!	Worktable	L agrees with F.
13	S:	Sorry.	Worktable	S apologizes for lack of focus.

Table 10.2 (continued)

Line	Student	Utterance	Location	Researcher interpretation
14	F:	Come on.	Worktable	F asks S to refocus.
15	S:	Okay, so what are we doing?	Worktable	S refocuses.
16	F:	Uh . . .	Worktable	F begins a verbalization.
17	L:	So it goes forward, turns, forward, turns when, when does it go wrong?	Worktable	L (looking at the computer screen) thinks aloud and moves her hands as if they were the robot moving across the table.
18	F:	I think it was that one.	Worktable	F (pointing at the screen) points at the block that needs to be programmed differently.

with strong supports for developing the ability to program and to think abstractly about the movement of the robot. Essentially, the 3-D activity of testing the executable program on the challenge board transforms student understanding of the 2-D programming icons. In this way, the three problem spaces work together to support learning about abstraction.

CREATIVE PROBLEM-SOLVING

In addition to supporting engagement in CT and science literacy practices, other modes of learning are strongly supported by robotics. These modes include play and creativity. Both of these modes of interaction support student engagement in problem-solving and learning with robotics. I argue that robotic devices are inherently playful; typically, the robotic device spurs student curiosity, and observing the movement of the device immediately raises a number of questions in students' minds about what the robot is and how it is doing what it does. Anecdotally, I have witnessed many students become intrigued with the device and express a desire to play with it; this desire to play with the robot serves as a means for learning more about it.

Playfulness can lead to resourcefulness when students are attempting to solve a robotics challenge. In a study conducted with students in a sixth-grade science classroom (Sullivan 2011), I used a Bakhtinian (Bakhtin 1986, 1981) lens to identify the reified and spoken voices that influenced students' collaborative development of a creative idea to solve a particular challenge. Integral to this analysis is the notion that the designed device itself embeds the intentions of the designers and affords certain types of interactions. Resnick (2003, 2006, 2014) has often discussed the role of play at the heart of the technologies he develops, such as the LEGO brick. This is in line with Papert's (1993) strong support for the idea of tinkering with technologies to learn more about them, but also to make them one's own. Moreover, the manipulative nature of the robotic device (i.e., one can hold it in one's hands), coupled with the fact that the device can be designed to roam around a room as a wheeled vehicle, affords a high degree of student interaction and provides an opportunity for students to think creatively about how to use the physical environment (the third problem space) to help them solve challenges.

In this particular study (Sullivan 2011), the students repurposed an item from the LEGO materials not used in the creation of the robotics device to help them solve the challenge. The repurposing of the item was an instance of bricolage (Lévi-Strauss 1966). Bricolage is the idea that one should use what is "ready-to-hand" to address current problems, regardless of the intended use of an object. This type of practice leads students to develop environmentally influenced problem-solving strategies and algorithms to solve robotics challenges.

In addition to creating environmentally influenced problem-solving strategies, we have also found that students developed strategies that entail the use of the device itself. For example, in a case study conducted with twelve students attending the three-week, 105-hour robotics camp referenced earlier in the chapter, we identified a problem-solving strategy we termed "simulating the movement of the robot" (Sullivan and Lin 2012). This strategy includes holding the robot (the first problem space) and moving it about the physical environment that constitutes the challenge space (the third problem space). We observed that, as students engaged in this activity, they often verbalized the program that needed to be written to solve the challenge. Here, one can recognize this activity from Vygotsky's

(1978) perspective as the role of externalized verbalizations and the use of tools in mediating student learning in the robotics environment.

Finally, in addition to engagement in problem solving, our research has indicated that students engage in a number of activities that emphasize computational concepts while working across the three problem spaces that make up the activity of learning with robotics. In our early work (Sullivan and Lin 2012), we examined the computational concepts that fifth-grade children engaged with while solving robotics challenge. For example, we have found that children had the opportunity to engage with conditional reasoning, program control and flow elements, and the basic idea of input/process/output. In our later work (Sullivan and Keith 2018; Sullivan, Söken, and Yildiz 2019), we developed a computational concepts coding scheme to assist in the analysis of student problem-solving conversations and activities across two different studies. In each of these studies we collected video data of focal student groups solving robotics challenges. We transcribed these data and analyzed student talk at the level of the utterance.

Our computational concepts coding scheme was both data driven and theoretically influenced from the literature (Barr and Stephenson 2011; Grover and Pea 2013; Wing 2006). The scheme includes five CT codes as follows: analysis, algorithmic thinking operations, algorithmic thinking variable, designing, and debugging. We split the algorithmic thinking code in two because of the relative sophistication of setting the variable parameter of a coding block (algorithmic thinking variable) versus simply selecting a coding block to use in the program (algorithmic thinking operation). In two different case studies, we observed students intensely involved in computational discussions regarding designing (problem space one), algorithmic thinking (problem space two), and analysis and debugging (problem spaces one, two, and three). Characteristic of student involvement was a relationship between the difficulty of the challenge attempted and the sophistication of the solution. In this way, we observed a phenomenon originally discussed by Dorst and Cross (2001) regarding the co-evolution of the problem definition and the designed solution; as students became more familiar with the problem spaces in which they were working, the more sophisticated the designed solutions became, both at the building level (problem space one) and the programming level (problem space two).

CONCLUSION

In summation, robotics is an integrated learning system comprising three interwoven, multidimensional problem spaces. Interaction within and among these problem spaces supports students' development of CT and their science inquiry abilities. A future research direction derived from our research is further investigation of the intersection of CT and disciplinary practices. As Lee et al. (2020) have pointed out, there are a number of newer areas of inquiry in STEM that blend computation and science: for example, computational biology. Future CT research should seek to further explicate the interdisciplinary relationships endemic to these new areas, such that powerful curriculum and pedagogical practices can be developed to support students' learning.

REFERENCES

Bakhtin, Mikhail Mikhailovich. 1981. *The Dialogic Imagination,* edited by Michael Holquist; translated by Caryl Emerson and Michael Holquist. Austin, TX: University of Texas Press.

Bakhtin, Mikhail Mikhailovich. 1986. "The Problem of Speech Genres." In *Speech Genres and Other Late Essays*, translated by Vern W. McGee, edited by Caryl Emerson and Michael Holquist, 60–102. Austin: University of Texas Press.

Barr, Valerie, and Chris Stephenson. 2011. "Bringing Computational Thinking to K–12: What Is Involved and What Is the Role of the Computer Science Education Community?" *ACM Inroads* 2 (1): 48–54.

Cooper, Tim. 2004. "A Systems-Based Robotics Curriculum." Unpublished master's thesis, New York: Teachers College, Columbia University.

Dewey, John. 1938/1997. *Experience and Education.* New York: Simon and Shuster.

Dorst, Kees, and Nigel Cross. 2001. "Creativity in the Design Process: Co-evolution of Problem–Solution." *Design Studies* 22 (5): 425–437.

FIRST LEGO League, 2011. *Food Factor*. http://www.firstlegoleague.org/sites/default/files/food-factor/food-factor-challenge.pdf.

Grover, Shuchi, and Roy Pea. 2013. "Computational Thinking in K–12: A Review of the State of the Field." *Educational Researcher* 42 (38): 38–43. https://doi.org/10.3102/0013189X12463051.

Hmelo, Cindy E., Douglas L. Holton, and Janet L. Kolodner. 2000. "Designing to Learn about Complex Systems." *The Journal of the Learning Sciences* 9 (3): 247–298.

International Society for Technology in Education and the Computer Science Teachers Association. 2011. *Operational Definition of Computational Thinking for K–12 Education*. http://www.iste.org/docs/ct-documents/computational-thinking-operational-definition-flyer.pdf?sfvrsn=2.

Lee, Irene, Shuchi Grover, Fred Martin, Sarita Pillai, and Joyce Malyn-Smith. 2020. "Computational Thinking from a Disciplinary Perspective: Integrating Computational Thinking in K–12 Science, Technology, Engineering, and Mathematics Education." *Journal of Science Education and Technology* 29, 1–8. https://doi.org/10.1007/s10956-019-09803-w.

LEGO Engineering. n.d. *LEGO Engineering*. http://www.legoengineering.com/alternativeprogramming-languages/.

Lévi-Strauss, Claude. 1966. *The Savage Mind*. Chicago: University of Chicago Press.

National Research Council. 1996. *The National Science Standards*. http://www.nap.edu/readingroom/books/nses.

Next Generation Science Standards. 2013. *Next Generation Science Standards*. https://www.nextgenscience.org/search-standards.

Papert, Seymour. 1993. *The Children's Machine: Rethinking School in the Age of the Computer*. New York: Basic Books.

Penner, David, E. 2000. "Cognition, Computers, and Synthetic Science: Building Knowledge and Meaning Through Modeling." *Review of Research in Education* 25: 1–35.

Resnick, Mitchell. 2003. "Playful Learning and Creative Societies." *Education Update* VIII (6).

Resnick, Mitchell. 2006. "Computer as Paintbrush: Technology, Play, and the Creative Society." In *Play = Learning: How Play Motivates and Enhances Children's Cognitive and Social-Emotional Growth*, edited by Dorothy Singer, Roberta Mitchnik Golikoff, and Kathy Hirsh-Pasek, 192–208. Oxford: Oxford University Press.

Resnick, Mitchell. 2014. "Give P's a Chance: Projects, Peers, Passion, Play." *Constructionism and Creativity Conference*, opening keynote speech, Vienna, Austria, August 19, 2014.

Resnick, Mitchell, Fred Martin, Richard Sargent, and Brian Silverman. 1996. "Programmable Bricks: Toys to Think With." *IBM Systems Journal* 35 (3&4): 443–452.

Roth, Wolf Michael. 2001. "Learning Science Through Technological Design." *Journal of Research in Science Teaching* 38: 768–790. https://doi.org/10.1002/tea.1031.

Scratch. n.d. *Scratch*. https://scratch.mit.edu/.

Sullivan, Florence, R. 2007. "Learning Through Building and Programming: Thinking and Reasoning with Robotics." Invited presentation, the STEM Education Institute, University of Massachusetts, Amherst, October 23, 2007.

Sullivan, Florence, R. 2008. “Robotics and Science Literacy: Thinking Skills, Science Process Skills, and Systems Understanding.” *Journal of Research in Science Teaching* 45 (3): 373–394.

Sullivan, Florence, R. 2011. “Serious and Playful Inquiry: Epistemological Aspects of Collaborative Creativity.” *Journal of Educational Technology and Society* 14 (1): 55–65.

Sullivan, Florence, R., and P. Kevin Keith. 2018. “Computational Thinking and Doing with Robotics in an All-Girl Workshop Setting.” Paper presentation at the annual meeting of the American Educational Research Association. New York, April 13–17, 2018.

Sullivan, Florence, R., and John Heffernan. 2016. “Robotic Construction Kits as Computational Manipulatives for Learning in the STEM Disciplines.” *Journal of Research on Technology in Education* 48 (2): 1–24.

Sullivan, Florence, R., and Xiaodong Lin. 2012. “The Ideal Science Student Survey: Exploring the Relationship of Students’ Perceptions to their Problem Solving Activity in a Robotics Context.” *Journal of Interactive Learning Research* 23 (3): 273–308.

Sullivan, Florence, R., Ali Söken, and Ozkan Yildiz. 2019. “Robotics and Science Inquiry: The Affordances of Sensors for Learning About Data.” Paper presentation at the annual Conference of the American Educational Research Association. April 5–9, 2019, Toronto, Ontario, Canada.

Vygotsky, Lev Semyonovitch. 1978. *Mind in Society: The Development of Higher Psychological Processes,* edited by Michael Cole, Vera John-Steiner, Sylvia Scribner, and Ellen Souberman. Cambridge: Harvard University Press.

Wing, Jean. 2006. “Computational Thinking.” *Communications of the ACM* 49 (3): 33–35.