Education and Information Technologies (2022) 27:3935–3963
https://doi.org/10.1007/s10639-021-10736-4

How do elementary school teachers learn coding and robotics? A case study of mediations and conflicts

Tugba Boz[1] · Martha Allexsaht-Snider[1]

Received: 14 May 2021 / Accepted: 30 August 2021 / Published online: 11 October 2021

Abstract
In this qualitative case study, we examined in-service elementary school teachers' learning of coding and robotics in a blended professional learning course developed and delivered through the collaboration between university faculty and a school district. We focused on activity theory to understand and reveal the mediations, conflicts, and effective practices that facilitated or hindered teachers' learning of coding and robotics. The participants of the study were twelve teachers from five different elementary schools in a rural school district. Data collection and generation sources included interviews, videos of class meetings, course assignments, and artifacts. In analyzing the data, we employed analytical approaches under the guidance of activity theory. The findings showed that teacher collaboration, coding/robotics platforms employed during the professional learning course, instructional approaches, and resources in and outside the professional learning setting mediated or conflicted with the teachers' learning of coding and robotics depending on the way that each of these elements was employed in the course. Elaborating on these elements, we reported the implications for further research and practice.

Keywords Coding · Robotics · Activity theory · Professional learning · Elementary school · Teacher education

1 Introduction

Developing coding skills and incorporating them in K through 12 is important for many reasons. First, coding is linked to self-expression, creativity, and innovation in STEM-related fields (Barr & Stephenson, 2011; Brennan & Resnick, 2012; DeSchryver & Yadav, 2015; Goode et al., 2012). Second, according to Goode et al., (2012, p. 47),

✉ Tugba Boz
tugba.boz@uga.edu

Martha Allexsaht-Snider
marthaas@uga.edu

[1] College of Education, University of Georgia, Athens, GA 30602, USA

today's students are surrounded by new technologies, but these students are mostly the "users" or "consumers" of these technologies, and "often only a narrow band of students that excludes significant numbers of girls and students of color" gain the opportunity as the "creators" of today's and future innovations. Therefore, coding in schools and providing more students with equitable opportunities and access to learning about it can increase the marginalized group representativeness in the creation of new technologies. Furthermore, according to a study by Master et al. (2017), coding leads to higher STEM motivation among first grade girls. Third, learning coding helps students with their development and use of self-regulation skills (Armstrong, 1989). In addition, coding encourages the development of metacognitive skills since children reflect on their own thinking as they solve problems on a coding environment (i.e., in debugging) (Clements & Nastasi, 1999). Finally, coding can be used to improve students' math skills through giving them spaces to apply their math skills (e.g., Barcelos et al., 2018), assess spatial skills (e.g., Bolognese, 2016), or teach physics (e.g., Lopez & Hernandez, 2015).

Given the significance of computing education in schools, some governments, such as England and Slovakia, have already included coding as a compulsory subject in their curricula starting from primary grades (Balanskat & Engelhardt, 2015). National policies and large-scale initiatives in other countries are currently under consideration. The United States have taken initiatives to encourage children to learn coding, and to expand and develop the computational practices and policies across the country (White House Archives, 2014, 2016). These policies and government agendas are also supported by many groups ranging from big software companies to data science hubs (e.g., Google, Microsoft, and Amazon).

It is anticipated that more teachers, especially in elementary schools, will be asked to teach coding quite soon. Surveys undertaken in different countries have shown that elementary teachers do not feel ready to teach coding to children (e.g., Gent, 2014; Ohashi et al., 2018). More studies on teachers' learning practices of coding are needed, as researchers currently do not have well-grounded answers regarding how to adequately prepare elementary teachers in teaching coding. Therefore, we conducted this research study to investigate the effective and meaningful ways to prepare current teachers for national and local initiatives which aim to integrate coding or coding and robotics within elementary school classrooms.

2 Background

2.1 Literature review on in-service elementary school teachers' learning of coding

Based on a collection of empirical studies, reviews, and conceptual frameworks, the literature clearly indicates that more research is needed to understand how to train current elementary school teachers in integrating coding and robotics into their classrooms. However, the current literature also shows that there are some emerging commonalities in terms of (a) teacher attitudes toward coding and robotics in elementary schools, (b) effective design and delivery of professional learning (PL) practices in coding and robotics, and (c) contextual factors to consider for such PL practices.

In terms of teacher attitudes toward coding and robotics, a common conclusion in most studies (e.g., Adler & Kim, 2018) was that even single day workshops positively impacted participants' attitudes toward coding. These studies also claimed that teachers could learn coding effectively, although they did not start off training with positive thoughts. For example, Ahern (2009) developed a one-week summer workshop. At the beginning of the workshop, the teachers were resistant to the idea that coding integration could benefit learning. However, their attitudes became more positive after the workshop instructors showed them how they could program a model in STARLogo and develop their own programs and models as instructional tools to include in their everyday lessons. Furthermore, most studies in the literature concluded that teacher confidence increased with increased familiarity with coding and robotics (e.g., Kay & Moss, 2012) and increased even more once teachers applied what they learned in their PL training and applied coding and robotics in their real life classrooms (Chalmers, 2018; Cooper et al., 2011; Holmes et al., 2018; Marcelino et al., 2018). Another commonly discussed conclusion was that teachers found the inclusion of coding and robotics fun and believed that the hands-on and collaborative nature of robotics increased their students' interest and motivation in class and developed their social skills (e.g., Savard & Highfield, 2015).

The researchers stated several implications and recommendations in terms of how to design and deliver coding and robotics training for current elementary school teachers. First, they stated that it was important to provide teachers with a context to go beyond computing and focus on what teachers could develop using coding, robotics, or open-source hardware (Ahern, 2009). In addition, Wolz et al. (2011) recommended that teachers needed to be asked to develop integrated lesson plans, but they also needed to focus on teaching the subject content more than on teaching the coding language (Scratch in their case).

According to some studies, teachers felt more comfortable when the workshop/training instructors walked them through writing a program in a stepwise manner and gave them unstructured time with options to work on (e.g., coding/robotics tasks of different levels of difficulty) (Kay & Moss, 2012; Liu et al., 2012). Although Sullivan and Moriarty (2009) used the discovery learning approach in their training "as a necessity of the continually changing face of technological innovation" (p. 134), they ultimately concluded in their study that their participant teachers found the discovery learning approach to be frustrating, consistently expressing interest in a more direct instruction for themselves and their students.

In addition, Dickes et al. (2016) indicated that there was a "reflexive relationship between computational thinking, scientific modeling and mathematical thinking when agent-based coding is the computational medium" (p. 9). Hestness et al. (2018) highlighted that computational thinking elements were already present in fifth grade science standards and could be acknowledged in order to facilitate teachers' acquisition of coding skills and their blending of these skills with their existing knowledge. For example, using the robotics kits, LEGO NXT and LEGO WeDo, Savard and Highfield (2015) claimed:

> *The mathematics involved in coding the robot is part of the mathematical context because it is all about using mathematics to code the robot to perform the task. There is mathematization or modelization of the situation. There are different processes involved and mathematical reasoning is absolutely necessary to code the robot in relation to the task to be performed. (p. 544)*

Along similar lines, Holmes et al. (2018) proposed that mathematics was necessary in teaching coding (Scratch in their case), and mathematical concepts needed to be made explicit in teaching coding.

Finally, in terms of logistics and contextual factors to be considered, Cooper et al. (2011) suggested that project developers needed to encourage technology teachers or media specialists to be involved in their training projects. They also suggested showcasing teacher work and seeking out community cooperation and partnerships. Collaborative work was emphasized in many studies as a contributing aspect of training events (e.g., Leonard et al., 2018). In addition, workshop designers were recommended include diverse materials and resources such as lesson plan samples and training materials (e.g., Cooper et al., 2011).

Overall, although we have some common perspectives and practices across studies and have some emerging answers to effective teacher PL in coding and robotics, we do not have a comprehensive study demonstrating how teachers learn coding and robotics and what facilitates or hinders their learning during the process. It is necessary that we should have more research studies in order to reach a consensus on presenting meaningful practices for in-service elementary teachers' robotics and coding learning and their such teaching and integration practices.

2.2 Theoretical framework: Activity theory

Activity Theory (AT, also known as Cultural Historical Activity Theory) seeks to understand human activities in their collective and social nature rather than privileging individual thinking or individual actions. AT as a theoretical framework was initially developed by Vygotsky and his successors as a reaction to the domain of psychology that was then dominated by the individualist and reductionist accounts of human development. Scholars investigating how people learn using AT theory begin their work with the assumption that "knowledge is commonly socially constructed, through collaborative efforts toward shared objectives or by dialogues and challenges brought about by differences in persons' perspectives" (Pea, 1993, p 48). Therefore, they examine learning and development as they occur when individuals interact with one another, other people, cultural artifacts, psychological (i.e., abstract devices that extends mental processes) and technical tools (i.e., physical objects such as books, materials, and computers). All these elements mediate the subjects' (agents of the activity system) realization of their socially shared object.

Engeström (1987) proposed that in an activity system, there are subjects (agents of the activity), object (or objective—shared goal of the agents), mediation (tools, signs, language which act as mediators between subjects and objects), division of labor and rules that mediate the activity, and community (agents who act in a wider social context in the system). For Engeström (1999), the focal element of the theory is the object(ive) since all other elements act as mediators in the system. In terms of the subjects of the activity, Engeström (1999) recommended exploring the supra-individual and social accounting for underlying tensions, conflicts, and contradictions. He claimed, "actions are not fully predictable, rational, and machine-like. The most well-planned and streamlined actions involve failures, disruptions, and unexpected innovations" (p. 32). In addition, individuals have agency and are free beings who pursue their own goals, and in doing so, they don't always follow existing norms. Therefore, it is harder to predict an individual person's actions than a social group's since individuals cannot necessarily "have their own norms and rules of activity that are only theirs. These norms and rules will always be shared by a number of people" (Lektorsky, 2009, p.79).

In this study, using activity theory allowed us to focus on processes that the teachers underwent in their learning journeys of coding and robotics. AT also made it possible for us to describe, narrate, and summarize a complex real-world learning environment into manageable and understandable units, hence, to evaluate the processes in this professional learning environment (Yamagata-Lynch, 2010).

3 Research design

We described the process of in-service elementary school teachers learning coding in a blended professional learning course (PL course) developed and delivered through a federally funded research project. The research questions of this qualitative study were:

How do elementary school teachers learn how to code in a professional learning course developed and delivered through the collaboration of university faculty with a school district?

What practices mediate or conflict with the process of these teachers' learning coding? What meaningful practices arise in the process of these teachers' learning how to code?

3.1 Research context

The research project was enacted through a nationally funded project which aimed to teach and support teachers (third to fifth grade) in integrating coding and robotics into their regular classroom practices using argumentation. Under this aim, the project team designed a professional learning course on coding and robotics for K-5 schools and taught it to the third through fifth grade teachers (including homeroom and specialist teachers) from the partnered school district, Somerville School District (all names are pseudonyms), in the southeastern United States. The course took place from January 2018 to May 2018. The authors of the paper did not design or teach the course. In this paper, we exclusively focused on the data of teachers' learning coding and robotics. A more detailed description of the professional learning course is presented below.

3.1.1 Coding platforms employed in the professional learning course

In the professional learning (PL) course, block-based visual programming languages and platforms were employed: Scratch, and two robotics kits, Ozobot Bits and RoboRobos, and their coding platforms, OzoBlockly and Rogic.

Scratch (https://scratch.mit.edu) is one of the most used coding platforms (available in more than 40 languages). It is designed for novice programmers and uses drag and drop block-based coding language. It is a trademark of Massachusetts Institute of Technology (MIT) Media Lab. Ozobots are walnut sized robots. OzoBlockly (https://ozobot.com/create/ozoblockly) is the platform that is used to code Ozobots and is powered by Google's Blockly. RoboRobo kits come with DC motors, sensors, LEDs, IR sensors, buzzer, and CPU board, and they can be built into various robot designs. RoboRobo (https://eng.roborobo.co.kr/main) uses Rogic, which is also a drag and drop visual coding platform. In the course, teachers with pre-course coding experience were also introduced to LEGO Mindstorms Education EV3 sets, which are comprehensive robotics sets that include three Servo Motors, five Sensors (Gyro, Ultrasonic, Color and 2 × Touch), and LEGO bricks. Sets can be used to build a variety of sophisticated robots.

3.1.2 Description of the professional learning course

The PL course had four face-to-face in-class meetings and five online coding assignments. The course's targeted learning outcome was that the participant teachers learned how to teach and include coding and robotics in their day-to-day lessons. The course instructor used whole group instruction, pair work, and groupwork throughout the course. He delivered the class materials through PPT presentations, handouts, and the use of in-class coding practices with robots. The course was designed and delivered to be interactive and provide ample opportunities for the participant teachers to collaborate with their peers and ask for support from the course instructor or other project staff (PI and co-PIs, research assistants, etc.). In addition, the course instructor provided feedback on course assignments and indicated the mistakes in each assignment the participant teachers submitted. The course instructor was a highly experienced faculty member. He actively taught coding and robotics to in-service and pre-service elementary and high school teachers.

As seen in Table 1, in the first face-to-face session, the teachers were introduced to coding and robotics, and the course instructor discussed the importance of coding and robotics for students' learning (i.e., to increase student motivation and engagement when integrated into other subjects in elementary schools) and real-world applications. Then, they learned coding terminology such as algorithm, pseudocode, control structures, and flowchart, and how these terms functioned and were identified within Scratch. Next, they were introduced to Scratch and asked to explore it for some time. At the end of the first class, they were also briefly introduced to

OzoBlockly and shown the code to *have Ozobots move on a square* activity. Before the second face-to-face session, the teachers turned in their first and second assignments. In their first assignment, they selected a concept to teach and create a corresponding Scratch program. In their second assignment, they were asked to complete the OzoBlockly Basic Training using ShapeTracer 1 and 2 (https://ozobot.com/create/games) and create two Ozobot programs. They also submitted a reflection about their experiences with ShapeTracer activity in OzoBlockly.

In the second face-to-face meeting, the course instructor elaborated on the control structures (sequential, repetitive, and conditional). He presented examples from OzoBlockly to teach control structures. He also asked them to reflect on control structures in mathematics and science disciplines. Then, he introduced RoboRobo robotics kits and their components, such as motors, sensors, and CPU boards, in order to facilitate teachers' understanding of how the kits would be assembled. The teachers started to assemble the kits by following the manual provided to them. For the third assignment, the teachers were asked to complete three tasks with RoboRobos. These tasks were to program RoboRobo to travel (a) on a straight line of 18–24 inches, (b) in a square with 10–12 side inches, and (c) in a circular walled area using sensors.

In the third face-to-face meeting, the instructor asked the teachers to program their robots to travel on a shape similar to a quarter-sized circle. He showed the video of his RoboRobo executing this task. He also explained how to have the robot turn at 90 degrees (by changing the wheel speeds and directions for the turns and curves). He differently asked the participant teachers with pre-course coding experiences to explore the LEGO Mindstorms EV3 and its coding software. He also gave a short review of different robotics kits markets available in the market. Then, the teachers spent 30 min on their tasks. At the end of the activity, the instructor

Table 1 Professional learning course schedule and content

Dates	Content
First in-class meeting	Introduction to class and robotics, Introduction to Scratch and OzoBlockly
Assignment 1	Select a concept to teach and create a corresponding Scratch program with a brief explanation of how this program would be used in instruction
Assignment 2	Complete OzoBlockly Basic Training, create two Ozobot programs, and write a reflection about the activity
Second in-class meeting	Revision of the control structures with elaborated examples, Introduction to RoboRobo (such as motors, sensors, and CPU board) RoboRobo assembly and Rogic (ports, delays)
Assignment 3	Program RoboRobo to travel on a straight line of 18–24 inches, travel in a square with 10–12 side inches, travel in a circular walled area using sensors
Third in-class meeting	In-class task to program RoboRobos in a shape similar to a quarter size circle
Assignment 4	Four tasks to debug code written in Scratch
Fourth in-class meeting	Review of progress with coding and robotics OzoBlockly activities (individual and pair work)
Assignment 5	Program RoboRobo to move on a curved line

asked the teachers to tell their pairs or group members what they did with their tasks and why in one minute. After the third class, the teachers submitted Assignment 4, which consisted of four tasks to debug some code written in Scratch.

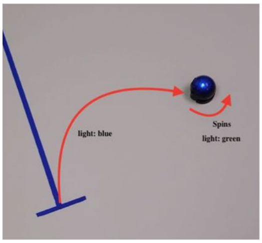
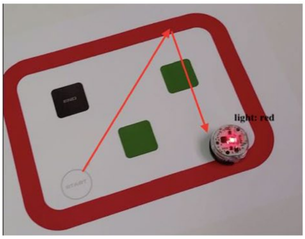
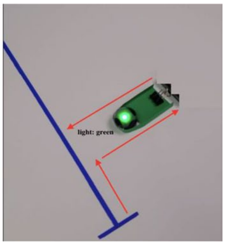
In the fourth face-to-face meeting, teachers were given a review of OzoBlockly, provided three videos of Ozobot (see Table 2 below), and asked to discuss what the robots were doing. After they discussed what the Ozobots in the videos were doing, they were asked to write and test the programs individually for their Ozobots to imitate the actions shown in the videos. Finally, they were asked to complete Assignment 5 which consisted of one task to be competed with RoboRobo having the robot make an S shape through manipulating the wheel speeds.

3.2 Participants

Fourteen teachers participated in the PL course in 2018. All teachers were employed in five different elementary schools out of six elementary schools in total in the partner school district, Somerville School District. Four of the participant teachers were fourth grade; one of them was third grade, and another one was fifth grade classroom teachers. Two of the teachers were media specialists. Three of them were Science Technology Engineering and Mathematics (STEM) specialists. One of the STEM teachers was also a gifted resource specialist, and one last teacher was an English as a second language (ESOL) teacher. Two of the participants were employed as administrators (assistant principals) in their schools. They were asked to join the course by their school principal and took the course to support their teachers. Therefore, they were excluded for the purposes of this research study. In total, twelve teachers compromised the participants of this research study (See Table 3).

Before the course, four teachers (Jessica, Bill, Hannah, and Sarah) had participated in the district's previous trainings for robotics and coding. They had more than two years of teaching experience with robotics and coding. They were confident when they began the course, and they enrolled in the course to learn about other robotics kits and coding platforms. Before the course, Beth and Doris taught coding through robotics to some extent. However, their experiences were limited. They enrolled in the course to learn about coding and robotics to teach them in their

Table 2 Fourth face-to-face meeting coding tasks and expected actions

	Task 1	Task 2	Task 3
Robot Actions in the			

current positions. The remaining teachers in the course indicated that they had no pre-course experience learning and teaching coding and/robotics, and neither did they have much idea about what coding would mean in an elementary school classroom. They either enrolled in the course voluntarily to learn about how to use coding to their homeroom students or they were asked to join the course by their principals.

3.3 Data sources

We generated data from the video recordings of the participant teachers during the face-to-face class meetings (see Appendix 1) and course assignments (see Table 1). We also used semi-structured interviews that the teachers undertook before and after they took the course.

3.4 Data analysis

Maxwell (2013) claimed that although "there is no cookbook or single correct way for doing qualitative analysis" (p. 105), it is important to plan the analysis in a way that fits the data at hand and guides the researcher in answering the research questions. In analyzing the data, we used different data analysis methods and techniques for each dataset. The first author of this paper undertook the coding of all data sources, and the second author supervised the process through cross checking the coding process across each data set: video data, pre- and post-course interview data,

Table 3 Participant data

Teachers	School	Grade	Years of teaching experience	Pre-course Experience with Coding and Robotics
Sarah	School 1	K-5 specialist (STEM)	16–20 years	Experienced
Beth	School 1	K-5 specialist (STEM)	16–20 years	Limited
Jessica	School 2	K-5 specialist (Media/Library)	11–15 years	Experienced
Dolly	School 2	3rd	3–5 years	None
Gloria	School 3	4th	6–8 years	None
Katy	School 3	5th	3–5 years	None
Whitney	School 4	3rd	More than 20 years	None
Lauren	School 4	4th	16–20 years	None
Doris	School 4	K-5 specialist (STEM)	11–15 years	Limited
Nadine	School 4	K-5 specialist (ESOL)	11–15 years	None
Bill	School 5	4th	9–10 years	Experienced
Hannah	School 5	K-5 specialist (Media/Library)	11–15 years	Experienced

Note. K-5 stands for K through Grade 5. STEM stands for Science, Technology, Engineering, and Mathematics. ESOL stands for English to the Speakers of Other Languages.

assignments, and memos. The authors reached the agreement on coding and analysis through cross checking within these multiple data sources employed in this study. Below we explicitly show how we analyzed each dataset.

3.4.1 Interview data

We employed three phases of coding and categorizing of pre- and post-course teacher interview data. In the first phase, we had a loose coding system and assigned data chunks into organizational categories that "function[ed] primarily as bins for sorting the data for further analysis" (Maxwell, 2013, p. 107). In doing so we looked for "resemblances or common features [across data] …independent of time and space" (p. 462). We had the following organizational codes in teacher interviews: (a) PL practice that mediated teacher's learning of coding, (b) PL practice that challenged teacher's learning of coding. These codes were substantive to our understanding of our participants' meanings and perspectives. As Maxwell (2013) indicated, these substantive codes were not conclusive but were primarily descriptive and did not hold any abstract ideas.

In the second phase, out of these substantive codes, we began developing analytical codes through looking for recurrent meanings, practices, and attributed roles and inquiring into the participants' articulated and unarticulated meanings and perspectives. Once the analytical codes were completed, we made sense of the analytical codes and incorporated them into analytical categories coming from other datasets.

3.4.2 Assignments

In analyzing assignments, we used two sets of substantive coding guidelines. The first set was for the written reflection tasks: (a) reflection on the task and (b) reflection on the experience. Based on these two codes, we developed the narrations of teachers' experiences with the assignments. The second set was to code the outputs coming from the coding tasks: (a) proficient, (b) developing with minor errors, and (c) developing with major errors. This part was to show teachers' progress and understanding of coding over the course.

3.4.3 Video analysis

According to Barron and Engle (2007), video data analysis should include both top-down and discovery-oriented approaches. In top-down approaches, video data is analyzed based on the theory guiding the research. Researchers focus on the theory-driven questions, which prevents them from getting lost in the vast amount of video data. In discovery-oriented approaches, researchers expect new discoveries and wish to uncover unforeseen occurrences. Using an inductive approach to analyze video data, we used discovery approaches and sampled and analyzed the video data simultaneously after we gained deep familiarity with the data. Therefore, data generation and analysis were undertaken together in this research. Barron and Engle (2007) recommended that researchers should engage in multi-stage phases of analysis since "an explicit multi-stage analytic approach can strengthen the likelihood of generating strong findings that are both reliable and valid" (p. 33). Erickson (2009) also recommended that sampling in video research should be conducted considering whether the instance chosen is a typical or an atypical instance, and how many similar or different instances are observed in the data. it would be significant to report "the full range of variation in occurrence of similar or analogous instances to show the reader that one has not cherry-picked evidence in reporting" (p. 73). Following these recommendations, we employed the systematic whole-to-part inductive approach and part-to-whole narrative evolving in sampling and analyzing video data in this study. For both approaches, videos were played at different speeds, slower or faster and with or without audio, to focus on phases of interests (Erickson, 1982).

We analyzed 16 h of video recording (see Appendix 1). The first author of the paper first watched each video recording completely and prepared content indexes. The second author supervised the process. Then re-watching the video, we asked the questions such as "within this class meeting, where and how often does event X happen? Does phenomenon X happen in every videotaped instance of this class meeting or only in some instances?" (Goldman et al., 2007, p.17). These questions also represented the form of thinking encouraged by grounded theory methodology (Glaser & Strauss, 1967) and Strauss' constant comparative method (1987) as well. Erickson (2006) defined such an approach to data generation and analysis as a whole-to-part inductive procedure. By employing this approach, we "strategically select[ed] events for analysis that adequately cover major themes and include key participants and hence constitute a kind of representative sample from the macro-event" (Goldman et al., 2007).

Then, out of the content indexes, the first author identified the recurring themes. Using activity systems analysis, the first author coded these themes as mediating or conflicting with teachers' learning of coding and robotics. Through narrative evolving, the first author described teachers' actions in the chosen video segments. The second author supervised the process of coding and narrative evolving. In reporting, we presented those moments from these recordings as excerpts (See Findings for those excerpts). Once the video data were analyzed, we started the last phase which was connecting video data with artifacts (course submissions of coding and robotics assignments) and interview data. We ensured the reliability and validity of the data analysis through data triangulation and cross-checking our claims across multiple data sources.

3.4.4 Using activity theory as the analytical lens

In this paper, we employed AT as our theoretical and analytical framework to reveal the elements that collectively mediated or conflicted with our participants' learning of coding and robotics (object—the shared goal of our agents) in the activity system of the professional learning course that we provided them. However, we did not separately focus on each of the AT elements. We only reported mediations, conflicts, and tensions related to the tools and signs of the targeted activity system (see Fig. 1). In delving into these mediations and conflicts related to the tools and signs of our targeted activity system, we aimed to understand how the elements of the

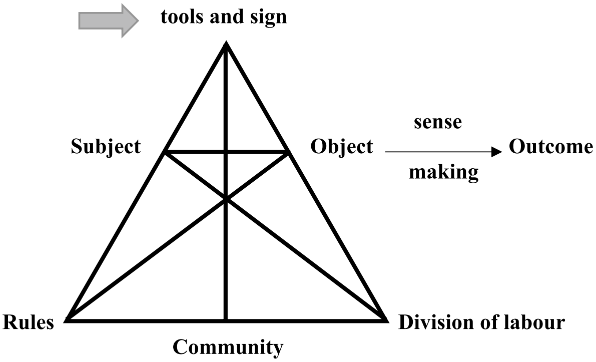


Fig. 1 Activity System (Engeström, 1987)

course facilitated or hindered teachers' learning of coding and robotics during the professional learning course. We reported the elements rules, community, and division of labor in another paper that would be published soon.

Using activity theory in this paper, we focused on the situational and contextual elements that occurred during the professional learning course. We examined these situational and contextual elements with the aim of revealing what mediated or conflicted with the object (shared goal of the agents/participants) of the activity (teachers' learning coding and robotics). In reading the transcriptions of the pre- and post-course teacher interviews, watching the videos, and reading the assignments, our analysis focused on revealing the mediations that enabled the subject (i.e., teachers in my study) to reach the object of the activity (i.e., teachers' learning of coding and robotics instruction), and challenges and conflicts that hindered the subjects' functioning in the investigated system (Yamagata-Lynch, 2003). To reveal these mediations and conflicts, we continuously zoomed in and out of the collective actions and meticulously paid attention to teachers' collective meaning-making practices and trajectories.

4 Findings

Based on the analysis of the course assignments uploaded online and the video recordings depicting teachers' engagement with the robotics/coding tasks during the face-to-face meetings, the following four major themes emerged as mediating or conflicting with participant teachers' learning of coding during the PL course.

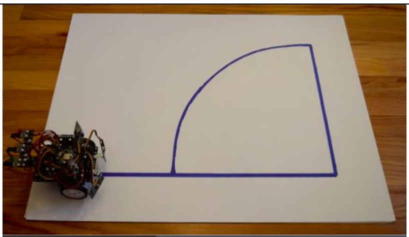
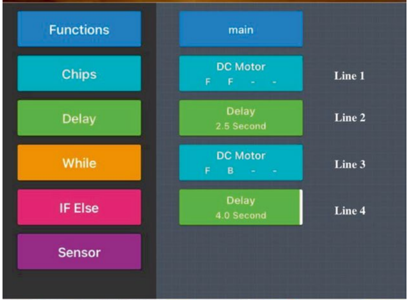
4.1 Collaboration

During the course, the teachers collaborated on completing the coding and robotics tasks. When they were assigned to do the tasks individually in the classroom, they still collaborated through asking questions to each other, looking at each other's screens and discussing their codes, and even sharing their concerns and receiving support from one another in forming lesson ideas. We identified four main collaborative practices in the teachers' group work interactions. Collaboration and groupwork facilitated teachers' learning of coding *(a) when the more experienced teachers (with pre-course coding experience) were grouped with the teachers with little to no experience of coding and/or robotics (or beginner-level teachers),* and *(b) when more experienced teachers were grouped together.* However, we observed that that a nurturing collaboration was not guaranteed in every instance of collaborative or groupwork interaction. Collaboration also conflicted with teachers' learning of coding *(c) when the experienced teachers were the majority in the group* and *(d) when the beginner-level teachers who made some gains throughout the course were grouped with the other beginner-level teachers who had made limited gains throughout the course*. Below we depict each type of collaborative interaction one by one.

To describe (a), when the experienced teachers were grouped with the teachers with little to no experience of coding and/or robotics (see Table 1 for participant information), the experienced teachers became the mentor and led the groupwork by equipping others with the necessary knowledge and skills to complete the class tasks. *To exemplify (a)*, in the second face-to-face meeting, Katy, Gloria, Sarah, and Beth were grouped together. The group expected to work on Rogic, *RoboRobo robot's coding platform* (details on robotics and coding platforms were discussed earlier in 3.1.1), to have their RoboRobo robot move on the given shape (see Excerpt 1 below).

The group programmed the RoboRobo to move on the first straight line of the given figure and make the first left turn. To make the RoboRobo travel the first straight line, they programmed both wheels to move Forward (See Line 1 in their program in Excerpt 1 below). Then, they added a *delay* block (*delay* is a timing block in Rogic; it is used to set the duration during which wheels will run; See Line 2 in their program in Excerpt 1 below). Then, they inserted another DC motor (See Line 3 in their program in Excerpt 1 below) to have the robot make the turn. In doing so, they set the left motor to be Forward and the right motor to be Backward (wheels with reversed directions make the robots turn). They set a delay of 4.00 s for the turn. However, when they ran their program, they encountered a problem. The robot moved on the straight line and then spun around itself in a clockwise direction (toward the right direction) instead of turning 90 degrees. Sarah was the first to understand from where the problems originated. The problem was the direction the wheels were turning in. The direction of the wheels needed to be reversed to make the robot turn left; the left motor needed to be Backward while the right motor needed to be Forward. The second problem was due to the duration in the second delay chip (See Line 4 in their program in Excerpt 1 below). For a 90-degree turn, they needed to decrease the duration (4.00 s) in the second delay chip because that duration made the robot turn 360 degrees instead. Sarah guided the whole group on how to solve these problems and even asked questions to the group to scaffold their understanding (see Excerpt 1):

Excerpt 1 Interaction among Sarah, Beth, Gloria, and Katy

Description	Content
Task	
Their program in Rogic	 main DC Motor F F - - Line 1 Delay 2.5 Second Line 2 DC Motor F B - - Line 3 Delay 4.0 Second Line 4
Picture	
Dialogue	Sarah: So, what happened with the motor [wheels]? Because that is concerning to me. *(Looking at the computer screen)* Aha! Beth: They are backwards, or? Sarah: No, they are not backwards. *(Holding the robot and showing the turns with the wheels)* We have programmed the left motor going forward and right motor going backwards. That means it is going this way *(turning right)*. (Gloria: Oh ok). So, we just got to switch that. *(Beth changes the directions of the motors/wheels in the DC motor chip in the third line of their program, which solved their first problem. Then, she attempts to change the speed—in the DC motor chip in Rogic.)* Sarah: Speed, or did we want to reduce the time? I think the time is easier to play with. Beth: Oh yes, where to…? Sarah: *(Beth tries to find a way to change the delay, and Sarah asks her to click on the little square on the delay block)* Put that [the mouse cursor] on the little square.

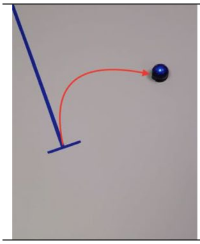
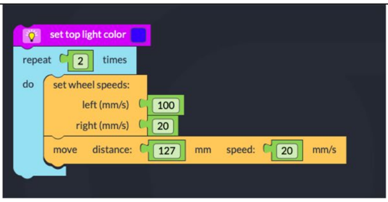
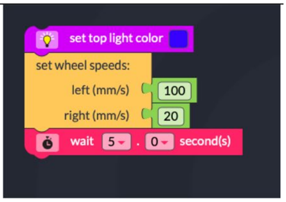
After they made the necessary changes to their program, their robot made the first 90-degree turn in the given shape. Since Gloria and Katy looked confused by the changes, Sarah asked Gloria and Katy whether they would like to change the speed on the program and gain experience. Gloria was not sure what to do when she first took the computer. However, Sarah guided her by showing Gloria the chips and telling her which chips to use. That was an important learning moment for Gloria; she began to understand the basics of the program by receiving first-hand guidance on coding as she learned to make the robot turn in the desired direction.

At the end of the activity, the course instructor asked the teachers to share with their partners (each other) what they did and why. Sarah asked Beth, Gloria, and Katy one by one how they programmed the robot, and even provided feedback on what they shared. Katy talked about how they programmed the straight lines. Gloria talked about how they needed to change the direction of the wheels to have the robot turn in the desired direction. Beth talked about how they programmed the 90-degree angles. All these gains in coding knowledge were built under Sarah's guidance throughout the activity. There was only one particular moment for Sarah to increase her coding knowledge. When Katy asked the group whether they could copy and paste the chips, Sarah indicated that she did not know, so she asked the course instructor. The course instructor told them that they could use the *functions* in Rogic. Sarah was fascinated by that option; however, she told the group, "it [function chip] is worth looking at. I mean we can take that out and trash them for now [do not need to play with it for now] … but that is, definitely, I am going to write that down." Although she took the note of the function chips, she could not elaborate on them, possibly because of her role as the guide in the group. She may have also thought that the function chip would be too advanced for the group to master at that moment. However, her contribution to the group helped all three teachers increase their coding knowledge.

To describe (b), when more experienced teachers were grouped together, they increased their coding knowledge and understanding through discussing their past experiences with various coding platforms. *To exemplify (b),* Jessica and Sarah were grouped to work on a task shown in Excerpt 2 below. Jessica was more familiar with Scratch than she was with OzoBlockly, while Sarah was more familiar with Rogic than Scratch. Therefore, when they needed to use the wheel speed block with Ozobots, Sarah did not encounter issues since the use of setting wheel speeds and duration was similar to the use of the wheel speed block in Rogic. Jessica, however, did have difficulty setting the wheel speeds because she confused *the wait block in Scratch* with *the wait block in OzoBlockly* (*wait* in Scratch is *stop for a certain amount of time* while *wait* in OzoBlockly is *duration* and similar to delay chip in Rogic).

As seen in the codes in Excerpt 2 below, Jessica dragged the motor block with speeds onto the coding canvas. She did not follow it with a wait (time) block but followed it with another movement block. Every time she ran her code; she was not able to get the robot to follow the desired path. She was sure that it was the robot that did not work properly. When she saw that Sarah could make her robot perform the targeted movement, she asked Sarah how Sarah accomplished this goal, as seen in the Dialogue in Excerpt 2 below.

Excerpt 2 Jessica and Sarah's programs for the coding task in the fourth meeting

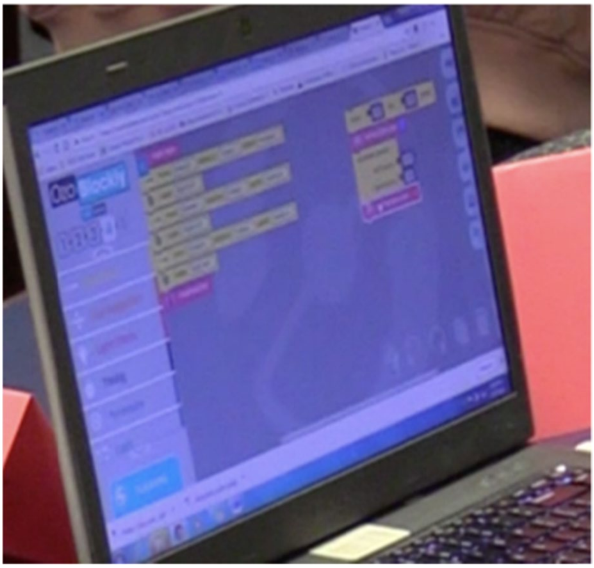
The Task	Jessica's Code	Sarah's Code
	 set top light color repeat 2 times do set wheel speeds: left (mm/s) 100 right (mm/s) 20 move distance: 127 mm speed: 20 mm/s	 set top light color set wheel speeds: left (mm/s) 100 right (mm/s) 20 wait 5 . 0 second(s)
Dialogue:	Jessica: How did you put the first part? Sarah: You have to put it [wait block]. You have to put it. Well, I call it a delay. 'Cause I am used to RoboRobo. Jessica: *(stands up and looks at Sarah's code on her computer screen)* What do you have the delay [wait in OzoBlockly] for? Sarah: 'Cause otherwise it doesn't know how long to do this *(showing the code: set wheel speeds)*. Jessica: Well, I wanted to do that the whole time while it's moving. But I think delay in this program [OzoBlockly] is actually like stop. Jessica: Did you just set the wheel speed and just a delay [wait in OzoBlockly], not the move forward, not anything? Sarah: No, I had the move [the code: set wheel speeds]. Jessica: That is the only thing that I can think of because it's … alright let me try this one. *(After trying out her new code with an inserted wait block and observing that her new code worked)* So it is just the delay. It is the delay.	

This excerpt shows that the design differences (i.e., the language used in the coding features of these platforms) meant that these teachers could not be sure of their coding and were confused by the different features of these coding platforms. Sarah was less confused compared to Jessica because Sarah reported that she was familiar with Rogic. However, Jessica confused the wait block in Scratch with the wait block in OzoBlockly because of the programs' shared word for different functions.

To describe (c), when the experienced teachers were the majority in the group, and there were others below their levels, the collaboration was mostly between the experienced teachers. The experienced teachers worked closely together to solve the task at hand. The teachers with less experience with coding/robotics tried to keep track of their thinking and actions. *To exemplify (c),* in the fourth face-to-face meeting, Beth was observed to be having difficulty with the in-class tasks. As mentioned earlier, the other two members of the group, Sarah and Jessica, worked on the difference of stop/delay blocks across the Rogic and Blockly programming. During that time, Beth tried to calibrate her Ozobot for 10–15 min on her own and then tested to see if her code worked. However, until the last minute of the activity period, she could not run her code. Right before Sarah left (she left 10 min before the class ended), Beth asked her and Jessica why her Ozobot would not run her code (see Fig. 2 and Fig. 3).

Then, Sarah showed her that she had two columns of codes. Beth should have removed one of them. That was an important and foundational learning moment

Fig. 2 Beth's code



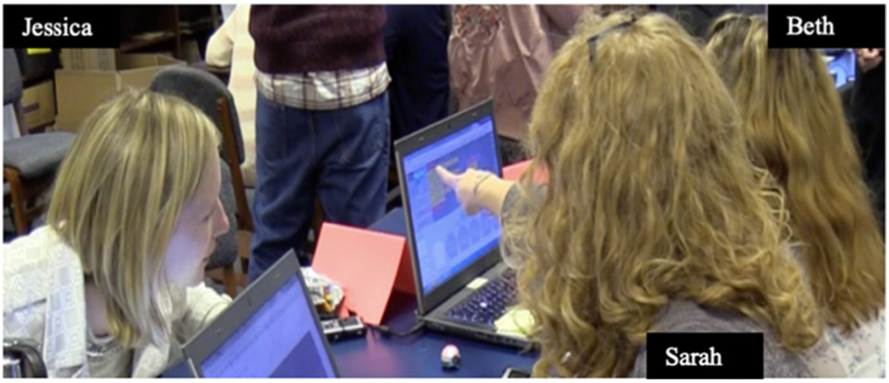


Fig. 3 Scene from the discussion among Beth, Jessica, and Sarah

for Beth. Until that moment, Beth had not gotten involved in the group discussion because Jessica and Sarah's level of discussion was outside of her immediate learning need and understanding.

To describe (d), collaboration also conflicted with the teachers' learning of coding (c) when the beginner-level teachers who made some gains throughout the course were grouped with the other beginner-level teachers who had limited gains over the course. When the teachers with lower levels of coding did not have any constructed contexts to refer to in coding the given tasks, they could not participate in their groupwork because group talk was not comprehensible to them. *To exemplify (d),* during the third in-class meeting, Whitney, Dolly, and Lauren were grouped together to complete the task in Excerpt 1 above and Fig. 4 below.

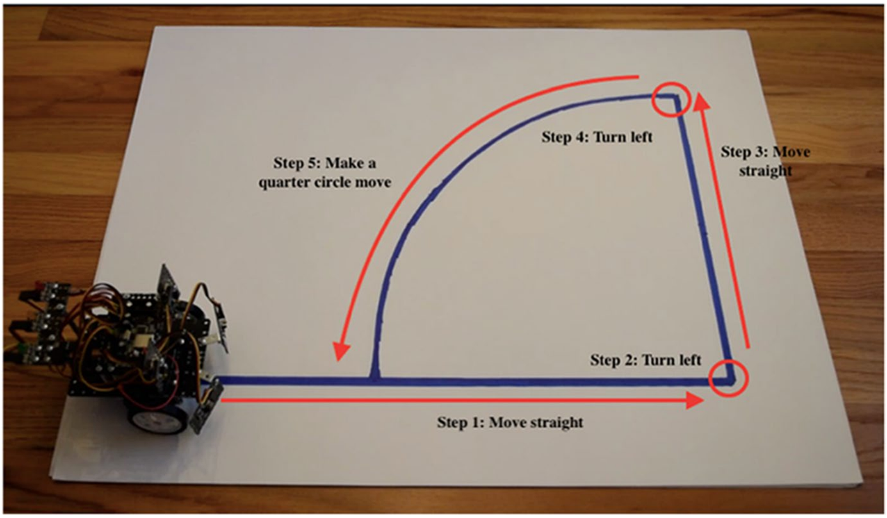


Fig. 4 Decomposing a task into pieces to program

Lauren and Dolly worked closely on the task, but Whitney could not get deeply involved in this interaction. The teachers needed to decompose the given task into steps to be able to code each step as shown in Fig. 4 below. Lauren and Dolly skipped that step because they already knew that they needed to do the coding algorithmically and step-by-step.

However, Whitney needed the time to decompose the task into pieces. Therefore, when the groupwork started, Lauren and Dolly began doing the work in Rogic. The discussion was already out of Whitney's familiarity, and the other two teachers were not aware of that. They did not do much in getting Whitney involved because they were fully engaged with the task themselves. Whitney could not get involved because the context was outside of her familiarity. A similar pattern occurred when Whitney worked with Doris in the fourth in-class meeting. Doris decomposed the task while guiding Whitney at the same time. During this process Whitney gained some familiarity with the algorithmic nature of coding. However, this time she was not familiar with how to translate the algorithms into blocks in OzoBlockly. Although she made effort to understand what and how Blockly worked as class time allowed, she could not understand what Doris was explaining to her about translating pieces of tasks into codes. Whitney eventually said, "I don't belong here." In both cases, Whitney excluded herself from the group (or pair work), and hence, the collaboration.

These four depictions of collaboration and groupwork types were the most observed interactions in the compiled video data throughout the course. Next, we describe how the use of multiple coding/robotics platforms influenced teachers' learning of coding and robotics.

4.2 Coding/Robotics platforms

The use of different platforms during the courses both mediated and conflicted with the teachers' learning and confidence, which was dependent on each teacher's progress in their acquisition of major computational control structures during the course.

Each coding platform in the PL course offered the teachers different built-in chips or blocks with specific features and syntax; however, all of these platforms allowed the teachers to manipulate *the same major computational control structures (sequential, repetitive, and selection control structures)* despite the differences in chips and blocks. During the courses, when the teachers gained the skills to develop these major computational structures in one platform, they gained flexibility and developed their capacities to transfer their skills to code in multiple platforms. Therefore, the use of more than one platform mediated and increased the teachers' understanding and application of coding. For example, as seen in Excerpt 2 above, Jessica, who spent extensive time with coding Scratch, experienced some problems with coding on different platforms; however, those problems were related to the design differences across platforms (i.e., organization of the chips or blocks) rather than her actual coding capabilities.

However, when the teachers did not build their coding confidence and did not develop *major computational control structures* in one platform, they felt uneasy transferring their skills from one platform to another. For example, Katy had some experiences coding in Rogic. However, she was still a developing learner of coding in Rogic when she was asked to code in OzoBlockly in the fourth in-class meeting. She could not make the connections between these two platforms. For example, in OzoBlockly, she needed to use a repetitive control structure (loops, repeat blocks, etc.). Although she successfully used loops in Rogic in one of her previous assignments (see Fig. 5), she could not do so in OzoBlockly (see Fig. 6; she listed blocks/codes one after another in OzoBlockly, which was also evident when she ran her codes and was clearly captured in the camera recording their groupwork).

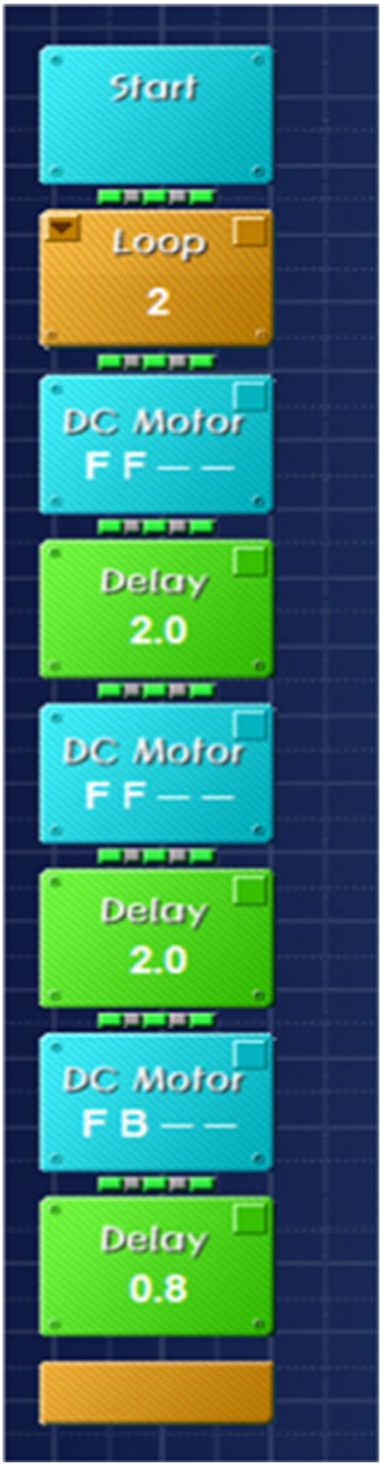
When asked in her post-course interview what challenges Katy had during the course, she reported that she would feel more comfortable being instructed on each platform one by one rather than altogether since each platform operates differently:

> *Probably, there's so many different coding, like OzoBlockly, the RoboRobo, Scratch, whatever else we used; I had to get used to every site and like what those commands do. And like RoboRobo, it says, like, slow, fast, and for how long, and then OzoBlockly is different. So, just getting used to all the different blocks.*

Other teachers likewise indicated that they would prefer learning and mastering coding in one platform truly before moving on to another coding platform. For example, Lauren said that she would be comfortable focusing on one platform at a time:

> *I think I would want to focus more on one particular format, so if it's Scratch versus Ozobot versus RoboRobo, I think that I need to, for me, the type of*

Fig. 5 Katy's submission for a part of Assignment 3 (March 27, 2018)



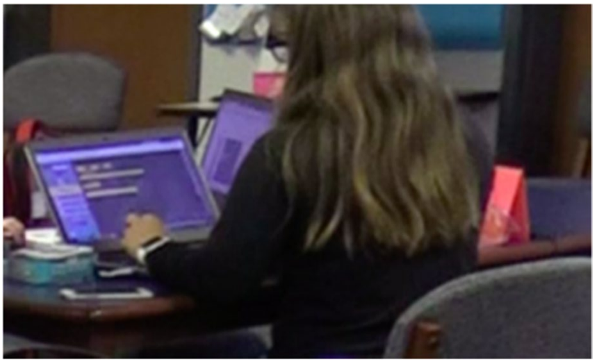


Fig. 6 Katy in the Fourth in-class Meeting (April 17, 2018)

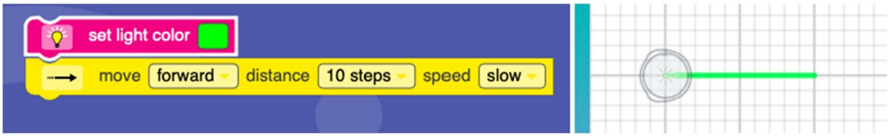
> *learner I am, I think I would like to become more proficient in one before I then try to put it all together.*

It seems that use of more than one coding and robotics platform during the course led to different results for different teachers based on their earlier pre-course experiences with coding and robotics in their positions and schools and their progresses throughout the course.

4.3 Instructional approaches: Structured or tinkering

In the PL course, we observed that the teachers benefitted from some degree of unstructured or free time to explore coding and robotics on their own. Most of the time, however, they preferred guidance and structure. For example, teachers reported positive experiences with the structured, built-in tutorials that helped them gain familiarity with the coding languages. For their first assignment in Scratch, Lauren said, “I found it interesting to go to some of the tutorials [in Scratch] and try to build something, that I might be able to, even if it’s not teaching coding, using what I made within my classroom.”

Furthermore, tinkering seemed to challenge teachers who were new to coding and robotics. For example, for their second assignment, the teachers were expected to explore and complete the ShapeTracer activity in OzoBlockly (See Fig. 7) The PL course instructor briefly introduced OzoBlockly to the teachers during the first class meeting and gave the second assignment to invite them to tinker and explore this activity in OzoBlockly. However, almost all the teachers (except for the experienced ones) reported difficulties with Assignment 2. For example, Doris, in her reflected on this assignment, claimed that she became “very frustrated.” She encountered



```
set light color 🟩
move forward distance 10 steps speed slow
```

Fig. 7 An example to OzoBlockly ShapeTracer

problem with the order of light chips and subsequently spent a great deal of time trying to resolve the issue. She stated that she was "VERY frustrated to finally figure out the problem was that the light had to go first instead of the movement!".

Another example to teacher reported difficulty came from Nadine, who had never encountered Rogic and RoboRobo before the third class meeting. Working on the task in Rogic for the first time, she asked for help from her peer, Doris, to help her identify which chips were needed to program the robot to move on a straight line. Doris showed her the fundamental chips (DC motor and delay) in Rogic. However, Doris could offer little support beyond this basic information because she was trying to complete her own class task. Therefore, Nadine, in turn, sought help from other colleagues and the course staff. She told everyone that she did not know where to begin and which chips to use. Nadine was only able to finally understand the process of coding in RoboRobo for her final assignment after consulting with the students in the robotics club at the school she taught in:

> *I was walking down the hall, and I saw the robotics team taking a picture, and I was like, "Oh! You guys could really help me with my final." Well, next thing you know, that*

afternoon, two of them who go to after school, showed up at my door. *Two of them [my students] who go to after school [robotics club], showed up at my door. "Well, what can we help you with?" I was like, "Oh, I don't even know where to start," and so they taught the vocabulary, they taught me the chips and what it means FF and FB and how I could make it, what I needed to do to curve this way.*

Therefore, the teachers preferred structure and guidance rather than free exploration and tinkering. Furthermore, when the teachers were new to coding, or when they were introduced to a new platform, they did not benefit greatly from listening to the whole group instruction because they did not have any reference points or constructed contexts to make sense of the new information. For example, in the second class meeting, the instructor introduced the whole group to the chips in Rogic, and told that one would need to manipulate wheel speeds and directions to have RoboRobos turn. In the beginning of the third class meeting, the instructor explained and emphasized again that the teachers needed to manipulate a robots' wheel speeds for turns and curves in Rogic. When the teachers started working on the task using Rogic for the first time, seven teachers out of twelve lacked any understanding of how to alter the robot's wheel speed to accommodate turns and asked for help from either their group colleagues or course instructors. For example, Lauren started coding; however, she did not know how to code to have her robot turn left. She immediately raised her hand to ask for help. One of the CALC staff came to help her and asked whether she heard what the course instructor said about the turns in his introduction to the task. Lauren said, "Yes, but I don't know [how to do it in Rogic]." Lauren also added that it was her first time using Rogic. The CALC staff then showed Lauren how to change the wheel speeds in Rogic for the turns. Whole group instruction seemed to mediate teachers' learning when the teachers followed the instruction while working on the platform first-hand and were being guided through the platform step by step. For example, in the first class meeting, the instructor introduced Scratch and guided the teachers through a Scratch program. Most teachers later responded positively to their learning experience with Scratch. Four teachers (Katy, Doris, Beth, and Gloria) even added that they introduced Scratch to their students right after that class. For example, Beth stated that she explored Scratch and Ozobot in greater depth at home after the first class. She described that experience as "immediate turnover of what we're learning and application in our arena."

4.4 Resources

Throughout the course, the teachers had access to various resources and materials available in the course's online learning management system. They had access to presentation slides, manuals/materials related to the robotics kits used in the classes, some instructional videos introducing the chips/blocks, and particular coding structures in the platforms employed. The course developers also provided additional handouts and booklets (materials that come with the kits). However, the teachers did not use the provided materials effectively unless they were asked to do so, or if the course instructor guided the teachers and showed them how they could use the materials and benefit from them. For example, there were introductory videos of Rogic and RoboRobos available in the learning management system. It is most likely that the teachers were not aware of those or did not know what to do with them, because the teachers looked for solutions on their own and occasionally had their progress impeded by coding questions that could easily have been answered by viewing those instructional videos. Instead, teachers looked for their answers to their questions in resources that they found outside the class. The teachers posed questions to people whom they thought might have knowledge related to their issues or who had experience teaching coding/robotics, such as the teachers of STEM classes and robotics clubs in their schools, media specialists, or the others who had participated in robotics training courses earlier. Some teachers listed online sources such as YouTube videos, built-in tutorials on the employed platforms (if any; for example, Rogic does not have built-in tutorials but provides links to some YouTube videos uploaded by the parent company), and lesson plans created by the companies of the robotics kits/online coding platforms or websites such as teacherspayteachers.com. For example, among the beginner-level teachers, Katy was the only one who completed Assignment 3 on time despite the fact that she reported challenges working with RoboRobo/Rogic. She claimed to have received help both from her colleagues and by looking up relevant information online:

> *After I had the software and programmed the robot to do the first task [using her school's media specialist's computer], it wouldn't run. That was frustrating. I called two of my friends who have RoboRobo backgrounds, and they helped me through it. I even watched some YouTube videos on how to make my robot go in a square. I finished tasks 1 and 2 at my house but waited until the next day to figure out task 3 with the STEM teacher at my school, Teacher Z. When I got my robot to run correctly, I couldn't help but let out a squeal!*

Most teachers, especially homeroom teachers, mentioned that they learned from their students or worked on their course assignments with their students who went to the robotics clubs in their schools. For example, Jessica, a teacher with substantial coding experience, similarly stated that sometimes she did not know the answers but some of her students did. She noted that she felt comfortable occupying the role of being a collaborator with her students rather than fulfilling the teacher role in which she should know the answers all the time:

> *I'm also not worried about the fact that the kids may figure out something before me or they might ask a question. I don't know the answer. But Johnny over here does. That doesn't bother me. Like, I'm not worried about that. I love that. So, but I can understand to where people may feel like, if I don't know, you know, what the kids asked me these questions, and I don't know how to answer it.*

5 Discussion and implications

In this qualitative case study, we examined in-service elementary school teachers' learning of coding and robotics in a professional learning course. Using activity theory, we analyzed the data to reveal the patterns of the mediations, conflicts, and effective practices that facilitated or hindered the teachers' learning of coding and robotics. During the course, collaborative actions, coding and robotics platforms, instructional approaches, and resources functioned as tools that mediated or conflicted with different teacher groups' learning of coding and robotics.

First, we observed in the PL course that teachers spontaneously collaborated with each other when they faced challenges in having their code work or when they needed to share what they did with their peers. However, this collaboration was nurturing when the people were grouped with the right people and could talk to each based on their progression in the course and within their familiar zone of coding and robotics knowledge and skills. As reported in the literature (e.g., Leonard et al., 2018) and observed in this study, in coding and robotics-related professional learning (PL) settings, teacher collaboration is an important component and naturally occurs due to the nature of the subject, coding—one's code can either work or cannot work; there is no in-between result. PL organizers need to ensure that teachers collaborate with the people they are grouped with to make their programs work and consistently build on their coding skills throughout PL courses. In doing so, teachers can communicate with each other and share a common language and skills on which they can ground and build their collaborative discussion and actions. Mere inclusion of collaborative acts is not enough to ensure that teachers are collaborating and indeed helping each other. It is crucial that the right supports be in place so that teachers do not fall behind or become discouraged as they negotiate individual and collective pathways of development for their learning and teaching of coding and robotics.

Second, the studies in this field (e.g., Chalmers, 2018; Holmes et al., 2018; Wolz et al., 2011) generally focused on teachers' training in a single coding language and environment. In this study's PL course, the teachers were introduced to many platforms because of the various resources (different robotics kits) available in the schools across the district and because the district used particular kits for the robotics competitions thus provided the STEM teachers with these kits. Some of these kits were cheaper compared to the other kits on the market. In addition, educational robotics technology has not advanced greatly and has yet to be developed in the industry. Generally, there are hardware problems (short shelf lives for some kits) and limited options to repair or replace broken pieces and devices. Therefore, users generally encounter many bugs and technical problems in developing their codes and running them. It is important for the educators and teachers to be aware of these issues and try out different options on the market to find the best ones for their contexts and students. This study's course was successful in introducing the teachers to different kit options on the market and the pros and cons of different kits so that the teachers could compare and choose the one that they would feel most comfortable with using in their lessons. However, the teachers needed more scaffolding and practical experience with one platform to be able to go beyond one context of coding. It seems clear that, depending on the teacher group's familiarity with the coding platforms and languages, one platform can be employed extensively in the beginning and more platforms can be introduced into the courses if deemed necessary by the administration or course developers.

Third, our participant teachers benefitted from direct instruction when they were already familiar with the platform being introduced during the whole class/direct instruction. When the introduced platform was completely new to the teachers, the teachers did not gain much from the instruction because they did not have established reference points or contexts to make sense of the new information. In providing direct instruction on coding and robotics to elementary school teachers, PL instructors can scaffold teachers' learning step-by-step throughout the whole-group instruction. PL instructors or designers can also provide teachers with many opportunities to build deep computational understanding. This can be realized through working on recognizing and composing algorithms and writing pseudocodes (bringing a complete and working program from bottom-up) and decomposing algorithms and a complete program/task/problem into individual elements. Teachers need to engage with such work in whole group instruction. Instructors can decompose some complete programs (or videos of robot's movements) into algorithms and show how each algorithm corresponds to certain blocks/chips in particular platforms. They can also ask teachers to talk through and plan their coding of the task at hand, and help them in their individual engagement with the algorithmic nature of coding. Teachers should be guided on how to start programming the task at hand to control their possible anxiety, and such guidance on how to begin a task is also important in leading the way for teachers to adopt the frame of mind necessary to complete a coding task.

Finally, the amount and variety of the materials presented to the teachers mediated their learning. However, the teachers did not use these materials unless they were asked to do so or particularly directed at using them. In finding the answers to their coding and robotics problems, our participant teachers collaborated with people outside the PL course setting and used many outside sources when they wrestled with the problems that they had in coding. Besides the provided materials, the teachers used YouTube videos, built-in tutorials of the platforms, lesson plans created by the companies of the robotics kits used, or websites including lesson plans on various subjects. It is important to plan and deliver PL programs that support teachers while they use multiple in and outside resources in coding, such as asking for help from peers in the course, their school colleagues with more specialized roles in teaching with technology, children in their school with interest and experience with coding and robotics, and online resources, or even the help desks of the vendors of the robotics kits. It is important for PL developers to systematically encourage teachers to identify and utilize these kinds of resources and for administrators to foster such practices in their schools as teachers collaborate with other teachers and children to learn coding and robotics themselves.

6 Limitations and future research

The study had a few limitations. In analyzing the video data, we had no control over the place of cameras, the use of panning and zooming, and the focus and framing of shots. On a few times, we had difficulties with the lack of zooming in because we could not clearly see some codes and screens. In addition, we struggled to capture the screen when the teacher blocked the shooting sitting right in front of the camera. However, sometimes there were two computers placed in the opposite directions which enabled us to see screens and the teachers' actions. Therefore, in future research, recordings of teachers' computer screens (or screen recordings) can be collected.

Further research can be conducted to examine the role of collaboration in teacher learning and teachers' perspectives on collaborating and being involved in groupwork with more experienced or with teachers who will be mentors in PL courses. Additionally, researchers can explore the pathways that teachers follow in engaging with resources outside teacher professional development settings. Furthermore, the elements found in this study under instructional approaches and delivery could be investigated in children's learning of coding and robotics. An important area to investigate could be the idea of children robotics and coding to their teachers or teachers' learning alongside children.

Appendix 1

Video Data List

The following is the list of the face-to-face video data watched and analyzed in this study (durations are rounded up to next whole unit):

Day 1 Instructor (50 min).
Day 2 Instructor (1 h).
Day 2 Dolly and Jessica (45 min).
Day 2 Gloria and Katy (45 min).
Day 2 Hannah, Bill, and Annie (45 min).
Day 2 Beth, Doris, Sarah (45 min).
Day 2 Nadine and Maureen (45 min).

Day 2 Lauren and Whitney (45 min).
Day 3 Instructor (30 min).
Day 3 Hannah, Bill, and Jessica (1 h).
Day 3 Sarah, Beth, Gloria, and Katy (1 h).
Day 3 Doris and Nadine (1 h).
Day 3 Lauren, Dolly, and Whitney (1 h).
Day 4 Instructor (30 min).
Day 4 Jessica, Sarah, and Beth (45 min).
Day 4 Nadine and Maureen (45 min).
Day 4 Lauren and Annie (45 min).
Day 4 Bill and Hannah (45 min).
Day 4 Doris and Whitney (45 min).
Day 4 Katy and Gloria (45 min).

Funding This work is supported by the National Science Foundation under grant number 1741910. Any opinions, findings, and conclusions in this paper are those of the authors and do not necessarily reflect the views of the National Science Foundation.

The study was reviewed and approved by the university's Institutional Review Board. Informed consents were obtained from all participants, and pseudonyms were used throughout the document.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

Adler, R., & Kim, H. (2018). Enhancing future K-8 teachers' computational thinking skills through modeling and simulations. *Education and Information Technologies, 23*(4), 1501–1514. https://doi.org/10.1007/s10639-017-9675-1

Ahern, T. C. (2009). Bridging the gap: Cognitive scaffolding to improve computer programming for middle school teachers. *Proceedings of the 39th frontiers in education conference (FIE)* (pp. 1–5). IEEE. https://doi.org/10.1109/FIE.2009.5350561.

Armstrong, A. (1989). The development of self-regulation skills through the modeling and structuring of computer programming. *ETR&D, 37*(2), 69–76. https://doi.org/10.1007/BF02298291

Balanskat, A., & Engelhardt, K. (2015). *Computing our future: Computer programming and coding priorities, school curricula and initiatives across Europe.* Retrieved April 3, 2019, from http://www.eun.org/documents/411753/817341/Computing+our+future_final_2015.pdf/d3780a64-1081-4488-8549-6033200e3c03

Barcelos, T., Muñoz-Soto, R., Villarroel, R., Merino, E., & Silveira, I. (2018). Mathematics learning through computational thinking activities: A systematic literature review. *Journal of Universal Computer Science, 24*(7), 815–845.

Barr, V., & Stephenson, C. (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. https://doi.org/10.1145/1929887.1929905

Barron, B., & Engle, R. A. (2007). Analyzing data derived from video records. In J. Derry (Ed.), *Guidelines for video research in education: Recommendations from an expert panel* (pp. 24–33). University of Chicago Data Research and Development Center. Retrieved August 21, 2019, from https://drdc.uchicago.edu/what/video-research-guidelines.pdf

Bolognese, C. A. (2016). Writing code to assess geometric reasoning. *Teaching Children Mathematics, 23*(4), 236–243. https://doi.org/10.5951/teacchilmath.23.4.0236

Brennan, K., & Resnick, M. (2012). *Using artifact-based interviews to study the development of computational thinking in interactive media design* [Paper presentation]. American Educational Research Association (AERA) Annual Meeting.

Chalmers, C. (2018). Robotics and computational thinking in primary school. *International Journal of Child-Computer Interaction, 17*, 93–100. https://doi.org/10.1016/j.ijcci.2018.06.005

Clements, D. H., & Nastasi, B. K. (1999). Metacognition, learning, and educational computer environments. *Information Technology in Childhood Education Annual*, 5–38.

Cooper, S., Dann, W., Lewis, D., Lawhead, P., Rodger, S., Schep, M., & Stalvey, R. (2011). A pre-college professional development program. In *Proceedings of the 16th Annual Joint Conference on Innovation and Technology in Computer Science Education (ITiCSE '11).* (pp. 188–192). Association for Computing Machinery Digital Library. https://doi.org/10.1145/1999747.1999801.

Deschryver, M. D., & Yadav, A. (2015). Creative and computational thinking in the context of new literacies: Working with teachers to scaffold complex technology-mediated to teaching and learning. *Journal of Technology and Teacher Education*, *23*(3), 411–431

Dickes, A., Farris, A., & Sengupta, P. (2016). Integrating agent-based programming with elementary science: The role of sociomathematical norms. In W. Chen, A. G. Barrera, H. Mitsuhara, G. Biswas, J. Jia, J.-C. Yanh, M. P. Banawan, M. Demirbilek, M. Gaydos, C.-P. Lin, J. G. Shon, S. Iyer, A. Gulz, C. Holden, G. Kessler, M. M. T. Rodrigo, P. Sengupta, P. Taalas, W. Chen,…, C. Yin (Eds.), *Proceedings of the 24th International Conference on Computers in Education* (ICCE) (pp. 129–138). Asia-Pacific Society for Computers in Education.

Engeström, Y. (1987). Learning by expanding: An activity-theoretical approach to developmental research. *Orienta-Konsultit*. https://doi.org/10.1017/CBO9781139814744

Engeström, Y. (1999). Activity theory and individual and social transformation. In Y. Engeström, R. Miettinen, & R.-L. Punamäki (Eds.), *Perspectives on activity theory* (pp.19–38). Cambridge University Press. https://doi.org/10.1017/CBO9780511812774.002.

Erickson, F. (1982). Audiovisual records as a primary data source. *Sociological Methods & Research, 11*(2), 213–232. https://doi.org/10.1177/0049124182011002008

Erickson, F. (2006). Studying side by side: Collaborative action ethnography in education research. In G. Spindler & L. Hammond (Eds.), *New horizons for ethnography in education* (pp. 235–257). Lawrence Erlbaum.

Erickson, F. (2009). Four points concerning policy-oriented qualitative research. In N. K. Denzin & M. D. Giardina (Eds.), *Qualitative inquiry and social justice: Toward a politics of hope* (pp. 73–80). Left Coast Press. https://doi.org/10.1177/1077800412454834.

Gent, E. (2014). *Most teachers not confident about teaching code.* E&T News. Retrieved January 15, 2019, from https://eandt.theiet.org/content/articles/2014/07/most-teachers-not-confident-about-teaching-code/

Glaser, B., & Strauss, A. (1967). *The discovery of grounded theory strategies for qualitative research.* Aldine Publishing Company.

Goldman, R., Erickson, F., Lemke, J., & Derry, J. (2007). Selection in video. In J. Derry (Ed.), *Guidelines for video research in education: Recommendations from an expert panel* (pp. 15–23). University of Chicago Data Research and Development Center. Retrieved April 3, 2019, from https://drdc.uchicago.edu/what/video-research-guidelines.pdf

Goode, J., Chapman, G., & Margolis, J. (2012). Beyond Curriculum. *ACM Inroads, 3*(2), 47–53. https://doi.org/10.1145/2189835.2189851

Hestness, E., Ketelhut, D. J., Mcginnis, R., & Plane, J. (2018). Professional knowledge building within an elementary teacher professional development experience on computational thinking in science education. *Journal of Technology and Teacher Education*, *26*(3), 411–435

Holmes, K., Hickmott, D., Prieto-Rodrigues, E., & Berger, N. (2018). Using coding to teach mathematics: Results of a pilot project. In *Proceedings of Integrated Education for Real World 5th International STEM in Education Conference* (pp. 152–158). Queensland University of Technology.

Kay, J. S., & Moss, J. G. (2012). Using robots to teach programming to K-12 teachers. In *Proceedings of Frontiers in Education Conference* (FIE) (pp. 1–6). IEEE. https://doi.org/10.1109/FIE.2012.6462375.

Lektorsky, V. A. (2009). Mediation as a means of collective activity. In A. Sannino, H. Daniels, & K. Gutierrez (Eds.), *Learning and expanding with activity theory* (pp. 75–87). Cambridge University Press. https://doi.org/10.1017/CBO9780511809989.

Leonard, J., Mitchell, M., Barnes-Johnson, J., Unertl, A., Outka-Hill, J., Robinson, R., & Hester-Croff, C. (2018). Preparing teachers to engage rural students in computational thinking through robotics, game design, and culturally responsive teaching. *Journal of Teacher Education, 69*(4), 386–407. https://doi.org/10.1177/0022487117732317

Liu, J., Lin, C. H., Hasson, E. P., & Barnett, Z. D. (2012). Computer science learning made interactive — A one-week Alice summer computing workshop for K-12 teachers. In *Proceedings of Frontiers in Education Conference* (FIE), (pp. 1–6). IEEE. https://doi.org/10.1109/FIE.2012.6462505.

Lopez, V., & Hernandez, M. I. (2015). Scratch as a computational modelling tool for teaching physics. *Physics Education, 50*(3), 310–316. https://doi.org/10.1088/0031-9120/50/3/310

Marcelino, M. J., Pessoa, T., Vieira, C., Salvador, T., & Mendes, A. J. (2018). Learning computational thinking and Scratch at distance. *Computers in Human Behavior, 80*, 470–477. https://doi.org/10.1016/j.chb.2017.09.025

Master, A., Cheryan, S., Moscatelli, A., & Meltzoff, A. N. (2017). Programming experience promotes higher STEM motivation among first-grade girls. *Journal of Experimental Child Psychology, 160*, 92–106. https://doi.org/10.1016/j.jecp.2017.03.013

Maxwell, J. A. (2013). *Qualitative research design: An interactive approach* (3rd ed.). Sage Publications, Inc.

Ohashi, Y., Yamachi, H., Kumeno, F., & Tsujimura, Y. (2018). Readiness of Japanese elementary school teachers to begin computer programming education. In *Proceedings of 2018 IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE)* (pp. 807–810). IEEE. https://doi.org/10.1109/TALE.2018.8615358.

Pea, R. (1993). Practices of distributed intelligence and designs for education. In G. Salomon (Ed.), *Distributed cognitions: Psychological and educational considerations* (pp. 47–87). Cambridge University Press

Savard, A., & Highfield, K. (2015). Teachers' talk about robotics: Where is the mathematics? In M. Marshman, V. Geiger, & A. Bennison (Eds.). *Proceedings of the 38th Annual Conference of the Mathematics Education Research Group of Australasia* (pp. 540–546). MERGA.

Strauss, A. L. (1987). *Qualitative analysis for social scientists.* Cambridge University Press. https://doi.org/10.1017/CBO9780511557842.

Sullivan, F. R., & Moriarty, M. A. (2009). Robotics and discovery learning: Pedagogical beliefs, teacher practice, and technology integration. *Journal of Technology and Teacher Education*, *17*(1), 109–142

White House Archives. (2014). *President Obama is the first president to write a line of code* [Press Release]. Retrieved June 18, 2018, from https://obamawhitehouse.archives.gov/blog/2014/12/10/president-obama-first-president-write-line-code

White House Archives. (2016). *Computer science for all* [Press Release]. Retrieved June 18, 2018, from https://obamawhitehouse.archives.gov/blog/2016/01/30/computer-science-all

Wolz, U., Stone, M., Pearson, K., Pulimood, S. M., & Switzer, M. (2011). Computational thinking and expository writing in the middle school. *ACM Transactions on Computing Education (TOCE), 11*(2), 1–22. https://doi.org/10.1145/1993069.1993073

Yamagata-Lynch, L. C. (2003). Using activity theory as an analytic lens for examining technology professional development in schools. *Mind, Culture, and Activity, 10*(2), 100–119. https://doi.org/10.1207/S1532-7884MCA1002_2

Yamagata-Lynch, L. C. (2010). *Activity systems analysis methods: Understanding complex learning environments.* Springer Science & Business Media.