Computer Science Education

ISSN: 0899-3408 (Print) 1744-5175 (Online) Journal homepage: https://www.tandfonline.com/loi/ncse20

Computational thinking in elementary classrooms: measuring teacher understanding of computational ideas for teaching science

Aman Yadav, Christina Krist, Jon Good & Elisa Nadire Caeli

COMPUTER SCIENCE EDUCATION
2018, VOL. 28, NO. 4, 371–400
https://doi.org/10.1080/08993408.2018.1560550

Routledge
Taylor & Francis Group

ARTICLE

Check for updates

Computational thinking in elementary classrooms: measuring teacher understanding of computational ideas for teaching science

Aman Yadav[a], Christina Krist[b], Jon Good[c] and Elisa Nadire Caeli[d]

[a]College of Education, Michigan State University, East Lansing, MI, USA; [b]College of Education, University of Illinois at Urbana-Champaign, Champaign, IL, USA; [c]Oakland Schools, Waterford, MI, USA; [d]Danish School of Education (DPU), Aarhus University, Copenhagen, NV, Denmark

ABSTRACT

A number of efforts have focused on preparing teachers to integrate CT within secondary disciplinary subject areas; however, there is little research on how CT ideas could be embedded within elementary subjects. We designed a professional development activity for elementary teachers to embed CT within science and examined how their understanding of CT emerged over the course of PD. This paper reports results from qualitative analysis of teacher responses to vignettes and open-ended questions, which presented teaching scenarios related to CT. We found that the vignettes allow us to see shift in teachersâ€™ thinking about CT from broad and generalized ideas to more elaborate versions of those ideas. We discuss that while vignettes provided a good method to portray changes in teacher views about CT, we need additional mechanisms to monitor how teachers conceptualize and come to integrate computational thinking into elementary schools.

ARTICLE HISTORY
Received 18 April 2018
Accepted 15 December 2018

KEYWORDS
Computational thinking; elementary; science

Introduction

The push to bring computer science in primary and secondary classrooms across the globe has led ministries of education, non-profits, and funding agencies to devote unprecedented resources to developing curricula and training teachers in computing. A majority of these efforts started out with a focus on coding as a vehicle to introduce computing in schools; however, more recently computational thinking has become a popular mechanism to bring computing ideas to classrooms. This shift from coding to computational thinking highlights the transdisciplinary nature of computing and how it can be leveraged to solve problems within disciplinary contexts. Wing (2006) characterized computational thinking as "solving problems, designing systems, and understanding human behavior by drawing on the concepts fundamental to computer science" (p. 33). This framing of computational thinking with ideas

CONTACT Aman Yadav ayadav@msu.edu College of Education, Michigan State University, Erickson Hall, 620 Farm Lane, East Lansing, MI 48824, USA

such as, problem decomposition, abstraction, and patterns provides teachers with a low threshold to bring computing to their classrooms and allows them to see connections between computational ideas and their subject area lessons.

One rationale for this approach is that it provides a strategic entry point for teaching computational ideas without needing to train teachers in computer science/programming (e.g. Weintrop et al., 2016). A number of researchers have suggested ways for teachers to leverage computational thinking in supporting their curricular and pedagogical needs within disciplinary content areas (See Barr & Stephenson, 2011; Weintrop et al., 2016; Yadav, Hong & Stephenson, 2016). For example, Weintrop et al. (2016) presented a taxonomy for thinking about the productive overlaps between CT and math and science instruction in order to support the infusion of CT into these disciplines. Barr and Stephenson (2011) not only suggested ways for CT integration in STEM area, but also provided approaches to bring computational thinking into non-STEM subject areas (such as social studies and language arts).

A second rationale is that computational thinking is not exclusive to computer science and therefore, may be useful for supporting student learning in other disciplines (e.g. Bundy, 2007). Emerging research has been promising in demonstrating benefits of computational thinking in improving student outcomes in other subject areas. For example, Calao, Moreno-León, Correa, and Robles (2015) showed that integrating computational thinking into sixth grade mathematics instruction led to gains in student understanding of mathematical processes. Sengupta, Kinnebrew, Basu, Biswas, and Clark (2013) also described the design of a computational learning environment that supported increase in students' physics and ecology learning. In parallel to the work on using computational thinking to support student outcomes, there is work emerging on how to engage teachers in these ideas. Krist et al. (2017) presented a case study of how primary school teachers themselves used computational thinking (CT) to support their own extended scientific inquiry. The results suggested that once teachers "explicitly identified a particular thinking strategy as 'computational thinking' as they used it as part of their science inquiry, they developed increasing ownership over this strategy (even naming it), making this element of CT a part of how they did science" (p. 4). This adult-level science inquiry was then used as a platform from which teachers could identify productive overlaps themselves and design science lessons that integrated CT for their students. Thus, applying computational thinking concepts and practices to other disciplines may be a productive pedagogical strategy, even without an explicit computer science instructional goal.

While there is work beginning to emerge on how to train teachers to integrate computational thinking within disciplinary contexts and its influence on students, we still do not have mechanisms to study changes in teachers' conceptions of computational thinking as they work to integrate CT into other subject areas. If we are to see how teachers take up computational thinking ideas, we need techniques that help gauge teacher understanding and how they conceive of CT within their classroom practices. In this study, we used teaching vignettes as a means to examine changes in teachers' views about computational thinking and how to integrate it in elementary science lessons over the course of a year-long professional development program. The following sections elaborate on computational thinking in primary and secondary curriculum, and current research on teachers' understanding of CT.

Computational thinking in curricula and standards

Computational thinking (CT) has a deep history with ambitious ideas being formulated since 1950s (Tedre & Denning, 2016). However, different paradigms have been favoured throughout time to define what CT involves. Denning and Freeman (2009) illustrated different attempts to provide unified views on computation, emphasizing either math, science or engineering traditions instead of supporting and accepting all three sub-paradigms as important. Wing (2006) re-introduced computational thinking (CT) as an essential 21st century skill that is on par with reading, writing, and arithmetic. She defined CT as "solving problems, designing systems, and understanding human behavior, by drawing on the concepts fundamental to computer science" (p. 33). In a review of computational thinking in K-12, Grover and Pea (2013) suggested a number of elements that form the basis of CT curricula: abstraction, systematic processing of information, symbol systems and representation, algorithms, problem decomposition, iteration, conditional logic, and debugging. Given that computation is becoming an integral part in today's technology-rich society, there is now a global push to introduce computational thinking ideas within primary and secondary education (Yadav, Good, Voogt, & Fisser, 2017). However, different initiatives have different views and conceptions of what CT involves.

A number of leading educational organizations, such as CSTA (Computer Science Teachers Association) and ISTE (International Society for Technology in Education) have also argued for the need to expose students to CT ideas so they are prepared to solve tomorrow's problems. For example, ISTE standards on computational thinking argued for the need for students to "develop and employ strategies for understanding and solving problems in ways that leverage the power of technological methods to develop and test solutions." In the United States, computational thinking is now included within the Next Generation Science Standards (NGSS Lead States, 2013), the most recent reforms aimed at improving science education in schools. In Denmark, the Ministry of Education in August 2017 introduced the optional experimental subject Teknologiforståelse (Technological Literacy) for lower secondary students. The subject involves teaching students to think computationally with a focus on i) designing and ii) programming digital products as the two main competence areas. The purpose is for students to develop an understanding of the possibilities of technology and the role it plays in society in order for them to be able to understand and act meaningful in a democratic digital society, hereunder being able to take part in shaping our digitalized world (EMU, 2017). In January 2018 the ministry announced more specific plans of making Technological Literacy mandatory for all children – not for all children to become IT Developers but for everyone to get a fundamental understanding of technology in our society (Aarhus University, 2017). Yadav et al. (2017) also presented cases from three countries – England, the United States of America, and the Netherlands – and how they were addressing the need for computational thinking in primary and secondary education.

A recent report by the European commission aimed to provide a comprehensive overview of CT skills for school children, including initiatives at the grassroots and policy level within Europe (Bocconi, Chioccariello, Dettori, Ferrari, & Engelhardt, 2016). The authors conducted an in-depth review of the literature, collected survey data from 17 Ministries of Education in Europe plus Israel and Turkey, and interviewed with 14 experts and policy makers from nine European countries. The survey results found that the main push to integrate CT in compulsory education was a need to foster logical thinking and problem-solving skills across all the countries. Some of the countries also saw the need to attract more students into computer science (N = 5) and foster coding skills as the rationale for CT (N = 7). The report also found that most countries had integrated CT at the secondary level and there was now a growing trend for integration at the primary level.

Denning (2017) argued that a consensus definition of computational thinking, how to assess it, and its universal value remain unresolved. He discussed that computational thinking has evolved over six decades, but the recent efforts to expand computer science in K-12 had led to computational tools that embodied a vague definition of CT in their designs. As a result, teachers and computer science education researchers struggle to answer: "What is computational thinking? How can it be assessed? Is it good for everyone?" (p. 33). An additional component central to integrating CT in K-12 education is that we still do not have a sense of how teachers come to conceptualize computational thinking and how it could be integrated within their day-to-day classroom activities. Accordingly, we add another question to Denning's list of unresolved issues: *How do teachers develop pedagogically contextualized understandings of CT?* The following section summarizes research done to examine teacher perceptions and understanding of computational thinking and motivates our approach for addressing this question.

Teacher understanding of CT

While there have been a number of initiatives on how computational thinking can be integrated within K-12 classrooms (Barr & Stephenson, 2011) and in teacher preparation (Yadav, Mayfield, Zhou, Hambrusch, & Korb, 2014; Yadav, Stephenson, & Hong, 2017), there is still limited research on how teachers come to understand computational thinking, especially in terms of how they come to think about embedding it in their own classrooms.

What teachers know (or not) about CT

Some existing research has used surveys to measure in-service teachers' perceptions of what computational thinking entails. For example, Sands, Yadav, and Good (2018) used a survey to examine inservice teachers' conceptions of computational thinking and how those conceptions differed based on teaching experience and subject matter taught. The authors found that while teachers agreed with statements that conceptualized CT as problem-solving, logical thinking, and algorithmic thinking, they also tended to view CT as mathematics, using a computer, and playing online games. There were no differences on how teachers viewed CT based on their background in STEM (Science, Technology, Engineering, and Mathematics) or grade level (primary vs secondary).

In another study, Blum and Cortina (2007) evaluated what teachers learned from a workshop designed to show the breadth and impact of computer science on other disciplines and their own lives. The authors found that teachers' conceptions of computer science evolved over the course of the workshop. Specifically, they shifted their view of computer science as being focused on problem solving/algorithms, learning/using programming languages, study of computation, and data organization to developing computational thinking skills, algorithms, use of computers to solve real world problems. This shift happened as teachers began to think of computer science of being more than programming. At the primary level, Corradini, Lodi, and Nardelli (2017) examined Italian primary teachers' conceptions about computational thinking. Based on a literature review, the authors first categorized CT elements into four categories: mental processes (such as, algorithmic thinking); methods (such as, automation); practices (such as, testing and debugging); and transversal skills (such as, creating). The analysis of survey responses from 972 teachers suggested that the vast majority of the teachers did not conceptualize computational thinking in terms of the four categories described above. Teachers also reported that they did not feel prepared to develop CT competencies in their students. One positive result from the survey was that majority of the teachers viewed computing and IT devices as separate things.

Similar work has been done at the preservice teacher education level examining how preservice teachers define computational thinking and its role in their future classrooms. Yadav et al. (2014) examined how a one-week module in a required educational psychology course influenced preservice teachers' understanding of computational thinking. The authors found that the module positively influenced preservice teachers' notions of computational thinking as drawing on computing ideas and practices to solve problems and they believed that they could implement CT ideas and practices in their future classrooms using connections to K-12 subject areas. Mouza and colleagues (2017) also examined how infusing CT in an educational technology course for preservice influenced their understanding and dispositions towards CT. The results suggested that the redesigned course increased preservice teachers' understanding of CT concepts and also positively impacted their beliefs on the value of using CT in their existing lesson plans. Results from these studies suggest that discipline-based integration that go beyond using programming can be a strategic means for introducing CT in K-12 schools.

In another study, Bower and Falkner (2015) examined preservice teachers' conceptions of computational thinking found that preservice teachers' descriptions of what CT was varied widely in sophistication and that being familiar with the term did not necessarily result in more sophisticated view of CT. When it came to what pedagogical strategies preservice teachers could use to develop students' CT capabilities, simple uses of technology emerged as the most popular response. Overall, the authors found that preservice teachers had a weak understanding of CT and majority had low confidence in their ability to teach CT.

Taken together, these studies suggest that teachers, without intervention, tend to equate CT with computing; and even for those who hold more nuanced views of CT, they express a lack of confidence in being able to meaningfully integrate CT into their classrooms. The studies also exhibited the potential of teachers to make broad shifts in what CT is and what it isn't. However, the kinds of pre/post tools that these studies used do not shed much light on *how* teachers come to think of these pieces together: how do teachers learn to think about what integrating CT into instruction would actually look like, in terms of the specifics of students' everyday activity?

Teacher learning about supporting students' CT: a situated lens

A situated understanding of CT–being able to imagine and/or identify what CT would look like in students' everyday activity–is important because developing and reflecting upon such understandings is the teacher learning that leads to meaningful shifts in instructional practice. Teachers develop an understanding of effective practice in any domain by experiencing concrete examples of teaching and learning, and reflecting on those experiences in order to make sense of them, and make decisions about the effectiveness of various practices based on these experiences (Horn, 2005; Korthagen, 2010). A critical part of this process is learning to notice particular features of interactions in practice, interpreting those features, recognizing which of them are potentially productive, and deciding how to respond to students' ideas (Barnhart & van Es, 2015; Santagata & Yeh, 2016; Van Es & Sherin, 2002). The causal reasoning involved in connecting what teachers notice about student thinking to deciding how to respond to it is an important mechanism by which teachers gradually change their practice (Dyer, 2016; Dyer & Kaliski, 2016).

Applying this perspective to the goal of integrating CT in K-12 classrooms implies that in addition to knowing *what* CT is, or what kinds of lesson activities might count as CT or not, teachers need to learn to notice the glimmers of computational thinking in students' everyday interactions and reflect on how those might be productive for developing deeper uses of CT. This noticing and reflection occurs both prospectively and retrospectively: how one should respond in the moment to a student idea or activity, and how that moment could be re-designed or re-framed to better support CT (Sherin, Jacobs, & Philipp, 2011).

Accordingly, we sought to develop methods of probing into teachers' thinking that allowed us to see what they were noticing about situated examples of classroom activity related to CT and how they were thinking about responding to or modifying instruction of that activity to improve the depth of students' CT. Most existing studies that do this use classroom video as the context for measuring/assessing teacher noticing (Sherin et al., 2011; Sherin & van Es, 2005). However, few (if any!) publicly available video clips of students doing meaningful CT embedded in the context of math or science existed at the start of this study. We chose instead to develop vignette-based prompts that asked teachers to reflect on a detailed description of student activity and interactions; to identify whether those students were "doing CT"; and to reflect on how they might respond to that student or modify the lesson in order to better support students in doing CT. This paper presents the design of these prompts and initial analyses of teacher responses at two time points during the first year of a three-year professional development program.

Noticing and supporting "unplugged" CT

In particular, we chose to focus on "unplugged" versions of CT. The idea that CT should not just be tied in the context of programming has been discussed for decades. Computer scientists and educators argued in the 1970s that preparing a program is a creative process (Knuth, 1974) and that programming is the least interesting part of a computational problem-solving process (Malmberg, 1970). For example, Malmberg (1970) emphasized the importance of learning how to express algorithms in ways a machine would be able to understand and not just when learning to program. He argued that the development of mental computational thought processes was an important aspect of thinking computationally. Similar, Knuth (1974) described computer science concepts as general-purpose mental tools that lead to much deeper understanding in any subject area and discussed how being able to express something as an algorithm prepares a person for much more than programming. Indeed, the post-Wing (2006) CT movement has defined computational thinking more broadly in order to expand access to computer science ideas for all learners, though this definition is not without controversy (see Denning, 2017).

Despite this emphasis on CT as mental tools that extend beyond computer-based environments, the majority of current professional development efforts have focused exposing teachers to programming environments rather than engaging teachers in thinking about how CT might be embedded it in authentic learning situations in other content areas (Sands et al., 2018; Yadav, et al., 2018). In response, we developed a professional development program that emphasized the latter. The vignette-based assessments that we present here are our attempt to capture how teachers were learning to notice, interpret, and respond to potentially productive glimmers of CT, as they were embedded in students' activity.

Aims of the current study

In this paper, we present the rationale for the design of the vignette-based assessment prompts as well as some initial analyses of teachers' responses before and after their first year of their involvement of a professional development context focused on integrating CT into elementary school math and science classrooms.

These assessments provided vignette-based descriptions of students' classroom activity and asked teachers to identify whether they saw CT in the vignette and how they might better support students' CT. This paper investigates the following questions:

- How can we design assessments that tap into teachers' situated understandings of and thinking about CT embedded in math and science instruction?
- Based on these assessments, how do elementary school teachers conceptualize computational thinking in practice, and (how) does their conceptualization of CT as it related to science inquiry shift over the course of an academic year?

Methods

Participants

Participants included nine teachers from two schools (Deep Creek and Parkside) in a large urban school district in the mid-Atlantic United States. The school district serves a diverse student population: 67.4% African-American, 22.6% Hispanic/Latino, 4.6% Caucasian, 2.9% Asian, and 2.4% other races. Parkside's student population reflected the demographics of the district while Deep Creek served a primarily Hispanic/Latino student population (98%). Four teachers from Deep Creek and five from Parkside elected to participate in the first three segments of our professional development project. These segments included a two-week long summer workshop in 2016, bi-weekly after-school sessions in their schools throughout the 2016–2017 academic year, and another two-week long summer workshop in 2017. These segments were delivered as a part of a funded project to integrate computational thinking and science inquiry in the elementary grades. During recruitment for the project, computational thinking was described as "the ways of thinking, or habits of mind, that computer scientists use" and that we as researchers thought that such habits of mind could be useful for science and/or math inquiry, though how exactly remained an open question. We asked teachers to come alongside us, drawing on their pedagogical expertise, to think about (a) what these computational thinking "habits of mind" were, (b) how they might be useful for students doing science and/or math, and (c) what instruction or activities might look like that engaged students in these computational "habits of mind" in support of science and/or math inquiry. Table 1 provides detailed information about the teachers in this study and their attendance for the initial summer workshop and 2016–2017 academic year.

Professional development context and CT instruction

The professional development workshops and after-school sessions for the study were designed from the following perspectives: that computational thinking skills and practices may support students' science and/or math learning; that "unplugged" approaches support students in developing CT, and in fact may better foreground the thinking strategies at play rather than the technological environments; and that teachers learn (about CT) through noticing, interpreting, reflecting upon, and making decisions about concrete examples of classroom

Table 1. Information about teachers.

Teacher	School	Grade Level and Subjects Taught	2016 Workshop Attendance (# days)	2016–2017 PD at school (# sessions)
Lynn	Deep Creek	4th grade science and social studies	8	11
Mila	Parkside	2nd grade (all subjects)	3	10
Travis	Deep Creek	6th grade math	9	11
Alyssa	Parkside	2nd grade	0	12
Irene	Deep Creek	K-6 math specialist	7	6
Ruby	Deep Creek	3rd grade science and math	9	11
Naomi	Parkside	3rd grade	0	10
Amy	Parkside	3rd grade science and math	5	11
Quinn	Parkside	5th grade science, social studies, and health	2	1

practice. These perspectives were implemented in various ways throughout each of the segments of the professional development project.

During Summer 2016, teachers participated in a 9-day workshop on integrating computational thinking into science inquiry. The general structure for the summer workshop included morning sessions that engaged teachers in adult-level science inquiry as learners and afternoon sessions that focused on pedagogical strategies for enacting inquiry-based instruction. Two days included sessions that explicitly introduced computational thinking concepts and skills. These sessions first engaged teachers in activities in which they themselves were using CT. Facilitators then led discussions reflecting on these activities and associating the thinking practices that the teachers identified with extant CT terminology (e.g. re-naming their description of revising a list of instructions when a partner could not follow them as "debugging"). Detailed descriptions of these activities are provided in Appendix A. As CT terms were collaboratively identified throughout the workshop, researchers identified ways teachers were using them during their morning adult-level science inquiry and encouraged teachers to intentionally draw upon them at key moments during their inquiry.

In addition, research team members (led by the second author) met biweekly with teachers at each school. These after-school PD meetings involved planning CT-integrated inquiry lessons and reporting and reflecting on those lessons. Four additional teachers joined the Deep Creek group and five additional teachers joined the Parkside group for these after-school meetings. Accordingly, two of these after-school meetings were sessions that paralleled the CT activities during the summer workshop. Based on these activities and the continuous lesson planning and reflection discussions, each group continued to collaboratively construct a working definition of CT and how it could be integrated into inquiry-based math and science instruction.

Measures: survey design and rationale

In order to measure how teachers thinking about CT emerged, we developed a survey based upon first author's prior work in measuring teachers' understanding of computational thinking (Yadav et al., 2014). The survey included two open-ended teaching vignettes to measure teachers' understanding of computational thinking in the context of teaching (see Table 2). Prior work has suggested that vignettes provide an opportunity to assess teachers' instructional practices, especially in the context of computing (Brovelli, Bölsterli, Rehm, & Wilhelm, 2014; Yadav, Berges, Sands, & Good, 2016). Brovelli and colleagues have argued that while traditional closed-ended assessments, such as multiple-choice tests ensure easy administration and stable psychometric properties, they could lead participants to pick socially desirable responses. Within the context of measuring teacher competencies, the authors also

Table 2. Teaching vignettes.

Vignette 1: Westwood Elementary school will start the next school year with a 1:1 iPad initiative. Mr. Nowak has decided to have his 2nd grade students use their iPads to predict weather (temperature, precipitation, and wind) for a week. Each student draws a picture of what they think the weather will look like. Sara, a student, also wanted to keep track of the temperatures that everyone predicted. Mr. Nowak started a Google spreadsheet where each student entered their predicted temperatures. The next day, they recorded the actual weather by using Accuweather App on their iPads and entering the information in the Google sheet. Olivia also wanted to record the actual temperature in Sara's spreadsheet so that they could compare how their predictions compared to what the weather actually was. After a week, they projected the Google spreadsheet on the smartboard and subtracted the differences between the observed and predicted temperatures. Mr. Nowak demonstrated how to make a bar graph of those differences.
Is Mr. Nowak engaging his students in computational thinking? In what ways are they doing computational thinking? In what ways they are not doing computational thinking?
Vignette 2: All the second-grade classes are taking a field trip! The school cafeteria packed PB&J lunches for everyone in identical paper bags, except for Sara and Olivia who have are allergic to peanuts. The lunch paper bags are labelled with all the student names and divided them up into 10 boxes with 10 lunches per box. The lunches were placed in boxes in alphabetical order by last name. Mr. Nowak wants to check to be sure that Sara and Olivia receive peanut-free lunches. They help him search through the boxes. Olivia Velazquez knows that her lunch will probably be near the end, so she looks at the first lunch in each box until she finds one starting with a letter close to the end of the alphabet. When she finds the box that begins with Jemal Summer's lunch, she then looks at the last lunch in that box. It is Billy Wagner's so she knows she must be close! She looks at the lunch right next to Billy's, and it is hers. Happily, she sees that the cafeteria remembered to pack her a cheese sandwich and carrots.
Is Olivia engaging in computational thinking? In what ways is she doing computational thinking? In what ways is she not doing computational thinking?

argued that such assessments are de-contextualized and do not reflect the complexity of teaching. In addition, they tend to "black-box" teachers' thinking processes in deciding upon their responses. Thus, we used text-based vignettes to provide a contextualized description of an instructional scenario. Given the exploratory nature of this work, we had teachers respond to open-ended prompts about the computational thinking portrayed in these vignettes, as such prompts have been shown to provide a valid form of assessment for measuring teachers' understanding of computational thinking (Brovelli et al., 2014).

The vignettes presented two contrasting scenarios with and without the use of technology to engage students in computational thinking. The first vignette focused on the use of technology (1:1 iPad, smartboard, Google sheets) to predict weather (temperature, precipitation, and wind) and compare how predictions matched with actual weather. Given prior research has suggested that teachers tend to focus on computational thinking as using computers (Yadav et al., 2016), we wanted to see whether and how teachers would see the use of technology tools in the first vignette as being necessary to computational thinking. The second vignette presented an "unplugged" scenario in which two second grade students went about finding their lunches from an identical set of paper bags arranged alphabetically in 10 boxes. This vignette aimed to see whether and to what degree of specificity teachers interpreted general problem-solving practices as part of CT. Teachers responded to open-ended prompts regarding whether and how students might be engaging in computational thinking in each of these vignettes. An additional question prompted teachers for their general understanding of computational thinking and how it may be used in conjunction with scientific inquiry practices in their classrooms. This question was intended to probe whether and how teachers were thinking about integrating CT into their own instruction.

Identical items were used on each administration. Though it was clear that teachers remembered these vignettes from pre to post (one Deep Creek teacher explicitly brought up one of them during an after-school discussion and compared it to an activity that another teacher was developing), the facilitators did not initiate discussion of the vignette situations during PD meetings. Facilitators were also careful to avoid simple binary classifications of something as either clearly "computational thinking" or clearly "not," instead encouraging identification of potential points of connection and exploration of ways to "strengthen" the CT component. Accordingly, in administering these same times pre and post, we were looking for deepened or more complex reasoning rather than shifts from incorrect to correct answers.

Data collection and analysis

Both before the summer 2016 workshop and after the 2016–2017 school year, teachers were contacted via email to complete the survey. Teachers were given multiple weeks to access the survey, along with time on the first day of the workshop for any remaining respondents to complete the survey. Prior to completing the 2016 survey, there had been no instructional sessions regarding computational thinking and any related topics. Prior to the 2017 survey, teachers had attended at least the 2016 summer workshop or bi-weekly professional development sessions, or both.

Upon completion of the 2017 survey, the responses were grouped by question and year for qualitative coding. We conducted an informed grounded analysis of the responses (Charmaz, 2006). We elected to take this approach to generate initial codes because we were looking for deepened, more elaborated, and/or more complex reasoning about computational thinking rather than simply the appearance or increase in frequency of a distinct computational thinking concept, skill, or practice (or a related disappearance of a CT "misconception"). A grounded analysis allowed us to capture this complexity in a way that remained true to the data. We, then, used theoretical sampling of literature (Thornberg, 2012) to identify existing frameworks for CT and leveraged core categories from an existing qualitatively-derived CT scheme (Good, Yadav, & Mishra, 2017) to categorize and collapse initial codes into themes. The goal was to use existing CT concepts from the literature in flexible ways and as heuristics tools that informed and helped focus our attention on how teachers conceptualized computational thinking (Thornberg). As Thorenberg suggested the theoretical sampling of literature helped us be more sensitive to the data by elaborating on our codes and comparing how they fit within the literature.

Two coders reviewed the initial codes to identify the prevalent themes that participants used when describing computational thinking. The resulting themes were iteratively revised in conversation with the data (Miles & Huberman, 1994) as well as Good et al.'s (2017) coding scheme. Once a stable set of themes emerged, they were collapsed into CT concepts three overarching categories: "CT is", "CT involves", and "CT Aids In". Codes were collapsed under "CT is" category when teacher responses focused around their definition of computational thinking. Similarly, when teachers discussed what practices and skills counted as doing computational thinking, they were put under "CT involves" category. Finally, "CT aids in" category contained codes that highlighted teachers' discussion regarding the types of practices that were supported by the use of CT. After this initial analysis, a third coder reviewed the entire coding data to resolve any disagreements between the initial two coders.

Results

Below we discuss the aspects of teachers' conceptions of computational thinking that the open-ended vignette-based assessments allowed us to capture. In addition, we provide some preliminary claims about how these ideas shifted over the course of teachers' first year of participation in a professional development project focused on integrating CT in their math and science inquiry lessons. Specifically, based on their reflections on these vignette-based scenarios, we discuss how we saw what teachers' definition of computational thinking (CT is), what they viewed as essential aspects of CT (CT involves), and what practices they thought could be supported by CT (CT aids in). These conceptions of CT reflect teachers' understandings of CT as situated in instructional contexts. Table 3 provides an overview of the overarching categories, corresponding themes, and examples resulting from the qualitative analysis.

CT is

Problem solving

Teachers at the beginning of the study often discussed computational thinking as being the same as problem solving. Data across the two vignettes and open-ended question showed that teachers responded to computational thinking as involving problem solving approaches before the 2016 workshop. These responses mainly focused around solving the problem presented in the vignettes. For example, one teacher Alyssa responded to the first vignette stating:

Table 3. Overarching categories, corresponding themes, and examples.

Category	Theme	Examples
CT IS	Problem Solving	"solving problems" "find a solution" "defining problems"
	Programming	"Students should be able to develop a program" "It could also require your child to write commands that a computer will be able to perform."
CT INVOLVES	Using logic	"The use of logic", "Her method showed an attempt to solve the problem logically using 'if then' types of thinking."
	Data Collection/ Analysis/ Representation	"…when they record the actual weather for that day…" "are also *compiling data* to test and refine their predictions" "gather, *chart*, and compare data about the weather"
	Algorithms	"a step by step logical method for answering her question" "Olivia is using an algorithm of sorts to find her lunch"
	Pattern recognition	"recognizing patterns", "students are looking for patterns in their predictions, patterns in the real world, and patterns that may exist between those two categories."
CT AIDS IN	Prediction	"Predicting and analyzing engages the students in computational thinking", "In order for this to be more like computational thinking, the students could use their data to develop a weather predictor system of some sort."
	Efficiency	"She is using computational thinking to find her lunch in the quickest manner possible" "Olivia is engaging in computational thinking. Olivia is using an algorithm of sorts to find her lunch in the most efficient way possible"

> I am not sure if Mr. Nowak is engaging his students in computational thinking or not. He is allowing them to make sense of their data collected by building spreadsheets, but the students are not solving a problem or using the computers in any way other than finding out the temperature.

Another teacher Amy had a similar response to the first vignette:

> I believe Mr. Nowak is engaging his students in computational thinking. His students incorporating technology and analyzing weather/mathematical patterns. It seems that students should be analyzing weather patterns in order to make their predictions, but it is unclear what information students used to make their predictions. It is unclear whether students are solving a problem or just making predictions. It is unclear if students are designing a way to make their predictions.

Teachers responded similarly to CT as problem solving for the second vignette as well, as highlighted by following comment from Amy.

> I believe Olivia is engaged in computational thinking. She is problem solving, and recognizing patterns

We saw significantly fewer responses (N = 3)[1] in the 2017 survey that were coded as CT being problem solving. While fewer in number, the responses showed teachers expanding on their ideas about problem solving by bringing in computational ideas, such as problem decomposition and algorithm efficiency, as highlighted by the following response from Travis:

> Yes, Olivia is using computational thinking to problem solve. I think what she is doing could be call chunking [problem decomposition]. She realizes that rather than looking at every lunch in every box, the boxes are chunked together by last name. She creates a system for quickly identifying what group of lunches are in each box and uses that system to find which box her lunch is in. Then, once she finds the box that has S – W in it, she searches only that box to find her lunch [efficiency].

Programming

In the 2016 survey, three teachers also discussed that computational thinking was connected to programming. The following response shows how teachers conceptualized relationship between computational thinking and programming initially:

> He is engaging them in computational thinking. The students could possibly be on their way to developing a computer program that will collect the data and display it in many different ways.

In another example, how computational thinking could play a role in science inquiry, a teacher mentioned:

> Students should be able to develop a program that will help them collect scientific data while engaging in a science inquiry activity.

Another teacher, Naomi, discussed the idea of programming in the context of using debugging approaches used when programming, stating:

> Olivia is using logical thinking skills, and she goes about finding her lunch with a strategy in place. She needs to find a box that starts with a letter towards the end of the alphabet. Using a logical strategy to sort or find something important seems to go with computational thinking. I don't know a lot about programming, but I think that sorting through to find an error or why something is not working in a systematic way would be an important skill to have. Olivia is just looking for her lunch, not doing any science or math. In that way, she is not applying computational thinking to a specific subject.

The connection between computational thinking and programming did not persist during the year and did not emerge in our analysis of responses from 2017 survey.

CT involves

We categorized a number of concepts and practices that teachers discussed within computational thinking under this broad category. Specifically, teachers discussed that CT involved using logic, data collection/analysis/representation, algorithms, debugging, plugged vs. unplugged activities. Below we discuss how teachers' ideas about these CT concepts/practices shifted over the course of a year.

Using logic

The idea that computational thinking involved logic emerged a number of times (N = 9) in teacher responses in the 2016 survey, but it stayed at a generic logical thinking level. For example, when responding to the second vignette, a teacher Lynn stated:

> So, this seems more like computational thinking. Olivia identifies a problem (does she have the correct lunch) and a solution (find her lunch and examine the contents) and then comes up with a step by step logical method for answering her question, i.e. start with a specific box, and depending on that result, proceed in a specific direction. This seems like a model of computational thinking even though she is not using a digital device.

Similarly, another teacher, Naomi, viewed computational thinking as involving logical thinking, stating:

> Olivia is using logical thinking skills, and she goes about finding her lunch with a strategy in place. She needs to find a box that starts with a letter towards the end of the alphabet. Using a logical strategy to sort or find something important seems to go with computational thinking. I don't know a lot about programming, but I think that sorting through to find an error or why something is not working in a systematic way would be an important skill to have.

The 2017 survey also had a number of responses that showed teachers' connecting computational thinking to using logic; however, their idea of logical thinking emerged to encompass conditional logic (if-then-else). The following response from Lynn for the same second vignette showcases this change:

> It does seem like Olivia is using computational thinking as she is developing a logical method of checking the lunches based on her knowledge that they are in alphabetical order. She might have made this more efficient by starting with the last box since her last name begins with a 'V' but the method she chose will lead her to her lunch with enough iterations, and probably more quickly than just randomly picking lunches. Her method showed an attempt to solve the problem logically using "if then" types of thinking.

Naomi also discussed conditional logic when responding to how computational thinking played a role in science inquiry, stating:

> I think students should be able to investigate their questions, and they should also have to record how they are going about their investigations. They should be able to do so in a specific way, like giving directions. They should be able to work towards developing if/then statements so that they can see cause/effect relationships.

Data collection/analysis/representation

A number of teacher responses (N = 25) showed that they connected data collection, analysis, and representation with computational thinking during the

2016 survey. For example, Mila when discussing how Mr. Nowak in the first vignette was engaging his students in computational thinking associated it with issues around data, stating:

> Yes, Mr. Nowak is engaging his students in computational thinking. There are several ways that computational thinking is being used. The students are using online databases and real-world observations to gather, chart, and compare data about the weather. This includes the use of the iPads to keep track of both individual student data and the class prediction chart as well was using an app to inform and check predictions. It is unclear whether the students are using technology to create their bar graphs, but I believe both a hand created graph and one made via technology count as computational thinking.

It is also interesting to note that Mila's responses connected computational thinking to the use of technology to collect, analyze, and represent data. Some teachers connected computational thinking to use of data when responding to the first vignette, but they were not always sure whether that was computational thinking as highlighted by the following comment from Alyssa:

> I am not sure if Mr. Nowak is engaging his students in computational thinking or not. He is allowing them to make sense of their data collected by building spreadsheets, but the students are not solving a problem or using the computers in any way other than finding out the temperature.

Teachers' ideas about how data was related to computational thinking continued to develop over the course of the year as showcased by Alyssa's response in 2017 survey:

> Mr. Nowak is starting to get his students thinking about computational thinking. He is allowing them to predict the weather based on what they already know and patterns that they see. They are also compiling data to test and refine their predictions. If they wanted to continue incorporating computational thinking they could try to create a list conditions in which each type of weather would occur.

Given the first vignette was written to examine how teachers would view the role of technology (iPads and Google Sheets) in computational thinking, we see how Alyssa went beyond simply using technology to how technology was being used to engage in computational ideas (i.e. pattern recognition using data). Furthermore, she saw the activity presented in the vignette as a starting point and suggested a new way to engage students in computational thinking – having students develop conditional logic for type of weather associated with a particular situation.

Algorithms

Teachers also discussed how computational thinking involved using/developing algorithms (step-by-step process) to solve problems. Similar to other aspects of teachers' thinking about computational thinking, teachers' ideas about algorithms also shifted from generic to specific over the course of the year. For example, Mila in her 2016 response to the second vignette responded:

> Yes, Olivia is engaging in computational thinking. She is using computational thinking to find her lunch in the quickest manner possible, by checking the first lunch in each box until she is close to her name. She was also using computational thinking by checking the last name in her selected box. This helped her to find her lunch quicker as opposed to searching through the whole box.

In this example, we can see that Mila brings up the idea of algorithms as a way for Olivia to find her lunch box and even goes as far as the speed/efficiency of the algorithm. When looking at her response to the vignette in 2017, we can see how she elaborates upon her ideas about algorithms and even brings the idea of abstraction of generalizing lessons from implementing the algorithm twice with Sarah to apply to find other peanut free lunches.

> Olivia is engaging in computational thinking. She is using an algorithm to sort through the lunches quicker. Knowing that the lunches are in alphabetical order and knowing that her last name is closer to the end of the alphabet, Olivia saves a lot of time in searching for her lunch by checking for boxes that are closer to the end of the alphabet, and then working backwards through that box. The computational thinking in this scenario could be taken up to another level is Olivia used the same method she had used to find her lunch, then help Sara find her lunch quicker as well. The two could discuss the pattern used and apply it to find the other peanut free lunch.

Some teachers' responses also showed connecting algorithms to playing a role in doing science inquiry as highlighted by Naomi in her response to the general question about computational thinking and science inquiry:

> I think students should be able to investigate their questions, and they should also have to record how they are going about their investigations. They should be able to do so in a specific way, like giving directions. They should be able to work towards developing if/then statements so that they can see cause/effect relationships.

Pattern recognition

Pattern recognition was another computational thinking concept that emerged in teacher responses. Given the two vignettes, it is interesting to note that pattern recognition emerged right away in the 2016 survey and did not change much over the course of the year as can be seen in Naomi's 2016 response to the first vignette:

> I think the students are coming up with strategies to observe patterns and record the patterns of the temperatures. In this way, it seems that Sara and Olivia are using computational thinking. The students are also learning how technology can help organize, compute, and display data. I am not sure if this is part of computational thinking or not.

In the above response, Naomi does not seem confident whether the use of technology is computational thinking or not and this concern persists and shows up in her response in 2017:

> Mr. Nowak is engaging his students in some computational thinking. The students seemed to come up with a logical system to keep track their predictions and the actual weather. It seems like at the end, the students are using an algorithm to find the difference in they predicted temperatures vs. the actual temperatures. The students made a bar graph of the differences, but I'm not sure if the bar graph they make will easily be used as far as computational thinking in concerned. Maybe the next step could be to use the actual weather data to look for patterns. Students could create some if/then statements to determine cause and effect relationships with elements of the weather.

When thinking about how computational thinking applied to scientific inquiry in the 2016 survey, responses focused on looking at data and patterns more generally as Alyssa stated:

> With my very limited knowledge of computational thinking, I would say that I see it fitting into science inquiry by having students categorize data, recognize patterns, and create new approaches to solving problems.

Alyssa expanded on how computational thinking plays a role in science inquiry by bringing in other CT ideas and practices as shown in her 2017 response:

> I think that computational thinking plays a role in science inquiry when a student is able create conditions for a scientific process and use those conditions to explain various things. For example, a student may use computational thinking to classify different information [patterning], to revise (or debug) their thinking about a scientific concept [debugging] or break down a problem into smaller more manageable pieces [problem decomposition]. I think that a student's ability to reason and revise outcomes is essential for both scientific inquiry and computational thinking.

CT aids in

We categorized responses under broader category "CT Aids In" when teachers discussed types of practices that were supported by the use of CT. Results showed that teachers' views about where CT could be helpful centered around two main themes: prediction, and efficiency.

Prediction

Teachers mentioned that CT could aid students in making predictions based on their observations/data and this view remained consistent across the academic year. For example, in her 2016 response to the first vignettes about Mr Nowak, Lynn responded:

> It sounds like the Internet is being used primarily to collect the actual weather results, not to help make predictions. My rudimentary concept of computational thinking is the use of logic to help make predictions or understand the cause of something – I don't see that here.

This view of how computational thinking can help make predictions remained for Lynn, but she was able to expand on her original idea, stating:

> But I associate computational thinking more in terms of the development of "if-then" scenarios and logical thinking. If they were trying to understand the influence of other factors on temperature and develop a method to predict what the temperature would be using those factors, that would seem more like computational thinking; essentially developing a model to predict temperature. Unless they developed reasons why they predicted the temperature as they did, it seems they were mostly just doing observations.

In his 2017 response, another teacher, Travis suggested that students in Mr. Nowak's class would be engaging in debugging if they could change their predictions based on observations. He wrote:

> One way to involve more computational thinking in this activity would be to have his students change their predictions throughout the week. For example, on Friday, they could make predictions for Monday through Friday of the following week. Then, on Monday, when they record the actual weather for that day, they could compare it to their prediction and change their predictions for the remainder of the week if needed. Then they could repeat this process on Tuesday and so on, all the way until Thursday. I think this would be sort of like debugging a program. They would be finding flaws and making changes as they "ran through the code."

Efficiency

Teachers also discussed how computational thinking could help achieve the task more efficiently and quickly. As discussed previously under Algorithms above, Mila's responses show how computational thinking helped Olivia before more efficient in finding her lunch box. The shift in how teachers viewed computational thinking as aiding in efficiency was evident from change in how Irene saw Olivia trying to find her lunch. In 2016, she stated:

> I don't believe that Olivia is engaging in computational thinking. he is using her knowledge of alphabetical order to locate her lunch.

While in 2017, she wrote:

> Olivia is engaged in computational thinking since she is using a "matrix" to find hers. She is using the order of the alphabet to efficiently find her bag rather than checking one bag at a time.

Discussion

The goal of this exploratory study was to examine teachers' conceptions of computational thinking and how those ideas emerged during the academic year as they engaged in unplugged approaches to integrating computational thinking in their classrooms. In addition, given that there are few ways to measure how teachers conceptualize computational thinking within disciplinary contexts, this study investigates teaching vignettes as a mechanism to observe and analyze changes in teachers' thinking about CT. In this section, we summarize shifts in teacher thinking our methods allowed us to observe, and what those shifts mean for educating teachers to integrate CT within elementary science classrooms. While the changes we observed were interesting within the context of our professional development program, we also discuss implications for researchers to go even further beyond black-boxed survey methods and better understand underlying pedagogical decisions that teachers take when implementing CT within their science lessons. In particular, we agree with Segall (2004) that CT ideas and lessons that teachers experienced during our professional development program were "not finished works of content awaiting pedagogical transformation; they [were], in and of themselves, pedagogical invitations for learning" (p.492).

Shifts in teacher thinking about CT in elementary science

Our results exhibited that teachers' ideas centered around what CT is, what CT involves, and how CT could aid in supporting other practices. We found that teachers had familiarity with computational thinking to begin with and their thoughts evolved over the course of the year. For example, when responding to the vignettes, a number of responses (N = 25) focused on data collection/analysis/representation in the 2016 survey; however, these were primarily focused around using technology to collect, analyze, and represent data in their responses to the first vignette (N = 15). Given that prior work has shown that teachers views about computational thinking initially tend of focus around use of technology, such as Excel spreadsheets (Yadav et al., 2014), it is not surprising that teachers in our study also focused on the use of iPads and Google spreadsheets when analyzing the first vignette. The connections to using technology as a way to engage students in computational thinking was less evident in 2017 survey as fewer responses (N = 8) to first vignette were coded under data collection/analysis/representation. While fewer in number, we found that teacher responses were more in-depth about how working with data was computational. This could be as a result of discussions during professional development and academic year being centered around unplugged computational thinking approaches, which showcased how there doesn't have to be technology present to thinking computational.

The teachers in this study showed greater familiarity with what CT is and what it involves than we expected, based on what has been reported for other K-12 teachers (Sands, Yadav, & Good, 2018). Some of this may be a selection effect: all of these teachers were the science and/or math specialists at their grade level, meaning that they taught science or math 2–3 times a day to rotating cohorts of students, rather than teaching every subject in a self-contained classroom. Thus, these teachers have likely had more science- and math-specific professional development than the average elementary school teacher. In addition, the state where the teachers taught was in the midst of a transition to aligning instruction to the Next Generation Science Standards (NGSS) in the United States, which includes computational thinking as a key practice. The state where teachers taught had adopted NGSS in 2013 and was in the midst of Phase 4, which included instructional shifts in primary and secondary classrooms to implement NGSS. The computational thinking practices from NGSS suggested that it included "range of tasks such as constructing simulations; statistically analyzing data; and recognizing, expressing, and applying quantitative relationships". Given the transition to NGSS in the state, it is not surprising that teachers had developed some ideas about computational thinking.

Even with this initial familiarity with CT, we see from the results that their conceptions continued to develop over the course of the year, especially with respect to how CT could play a role in science inquiry. We can describe the shifts reported above more generally as a shift from generalized, coarse-grained ideas (e.g. broadly defining CT as problem-solving) to more elaborated versions of these ideas, albeit in piecemeal ways. This is seen in the overall pattern that in 2017, they more often identified components of what "CT involves" rather than umbrella descriptions of what "CT is" as compared to 2016. More specifically, there were three concepts that became more specified over the course of the year: (1) the idea that CT involves logical thinking shifting to more specific references of one type of logical thinking, conditional logic; (2) the idea that working with data and CT are somehow related shifted to more specific suggestions of how students could continue to work with data in ways that could more substantively use CT, suggesting that teachers were seeing the presence of data as a potential starting point for CT; and (3) the idea that CT involves algorithms shifted to contain a more elaborated description of what the algorithms were, including tie-ins to abstraction, generalization, and/or how algorithms could be useful in the context of science.

Though the overall pattern of teachers' ideas of CT becoming more specified persisted across all participants, the particulars of which ideas became more specified varied between participants. This is likely due to the design of our PD, which aimed to develop CT concepts and skills *as they became relevant to teachers' activity and instruction*, rather than naming a set of concept and skills ahead of time and asking teachers to figure out how to incorporate them. As a result, we never provided an abstract definition or description that was not grounded in an experience that the teachers or their students had had. For example, we only named "iteration" as a CT practice after the teachers saw their students testing out a flow chart, finding a problem, fixing it, and testing it out again.

This approach likely directly influenced the shifts we saw in two ways. First, because of the unplugged context, we never identified programming as CT, because neither the teachers nor their students were ever programming. This likely explains the disappearance of mentions of CT as programming from pre to post. Similarly, we identified a lot use of conditional logic in teachers' own science inquiry, and several teachers incorporated conditional logic into their classroom lessons, again likely explaining the shift from general logical thinking to specifying conditional logic pre to post.

The second way that our approach likely influenced these shifts is more subtle. Our discussions of CT were embedded in practice and often involved conversations about how we could "amplify" the CT (make it more salient, or more central to the scientific or mathematical activity that was the focus of the lesson). Thus, in addition to naming particular CT concepts and skills, we were also asking teachers to engage in pedagogical thinking around how those concepts and skills might play out. A few teachers did this explicitly in their 2017 responses. Alyssa's response to the 2017 vignette (as presented above) captures this kind of thinking explicitly:

> Mr. Nowak is starting to get his students thinking about computational thinking. He is allowing them to predict the weather based on what they already know and patterns that they see. They are also compiling data to test and refine their predictions. If they wanted to continue incorporating computational thinking they could try to create a list of conditions in which each type of weather would occur.

Alyssa was able to identify the "seeds" of computational thinking that were already present in a fairly typical elementary science activity: predicting based on patterns, and using data to (presumably) test and refine predictions. More importantly, she was able to articulate a "next step" that would involve another aspect of computational thinking: creating a list of necessary conditions that could be used as a more formal prediction algorithm. Though responses with explicit pedagogical suggestions were not common, the kind of specification captured in shifts (2) and (3) reflect the mindset of seeing "seeds" of computational thinking as starting points for designing more CT-rich instruction. We argue that this kind of thinking is necessary for integrating CT into STEM activities in a meaningful, sustainable way.

Finally, we note that these teachers universally saw CT as something that could be productively integrated into doing science. In addition, they saw it as related to, but not the same as, some common components of science. For example, Alyssa highlighted collecting data and testing and refining ideas. These are practices central to doing science; they are also spaces where the teachers saw CT as useful to enhance the doing of science: to organize ideas such that it makes prediction easier or more powerful, to articulate if/then relationships to more clearly identify cause/effect relationships, or applying sets of rules or algorithms to multiple contexts to increase their generality (a process that is also central to the model-building work of science). In this way, it seems that teachers came to see CT not only as related to, but also a powerful addition to, the structuring and doing of science.

We do recognize that our approach conceptualizes CT as a tool for science, and therefore, likely only highlights the areas of CT that are easily overlapping (and therefore easily integrated). We are likely missing out on some of the power and breadth of CT and/or computer science more broadly. Although this approach to integration is not a stand-alone solution for teaching CT at the elementary level, we have shown here some important shifts in teacher thinking that make it more feasible, and therefore more likely, for teachers to begin integrating CT into their classrooms. We think this is a very promising starting point.

Given that the limited prior work on teachers' conceptualization of computational thinking has shown that teachers view it as a problem-solving approach or simply as mathematics and use of computers (Sands et al., 2018; Yadav et al., 2014), our results show a promising to engage teachers in productive shifts in integrating CT in elementary science. Furthermore, our finding is a step toward addressing Denning's (2017) concerns that teachers lack an understanding of computational thinking and how it might be good for everyone. Specifically, teachers in our study elaborated upon their initial CT conceptions and were able to make connections to how it plays a role in science inquiry. Our work also provide support on engaging teachers in Knuth's (1974) idea about computer science, which he suggested involve using general-purpose mental tools that lead to much deeper understanding in any subject area. As elementary teachers begin to integrate computational ideas and intentionally use computational terms in their classrooms, it could help demystify and broaden participation in computer science.

Beyond black-box assessment of teacher thinking

In this study, we used text-based teaching vignettes to measure changes in teacher thinking on integrating computational thinking in elementary science classrooms. Our findings suggest that this approach provides a good start to see how teachers conceptualize CT ideas, and their ideas about how CT could be a productive tool for facilitating science learning. The benefit of using text-based vignette was that it allowed us to go beyond the closed-ended assessments, which are de-contextualized from the complexities of classroom instruction. Additionally, it allowed us to measure how teachers made sense of computational thinking within the context of classroom instruction and their reasoning behind CT as a tool to teach science. This allowed for capturing of more subtle and nuanced shifts in teachers' conceptions of CT as they are relevant for reflecting upon and planning for integrating CT in mathematics and science instruction. In this sense, these vignette-based assessments "un-black-boxed" some degree of teacher thinking about CT.

As discussed previously, viewing and reflecting upon classroom video would provide the "gold standard" context for measuring teacher thinking (Sherin et al., 2011; Sherin & van Es, 2005). Video-based vignettes would ground the videos in classroom events even further to provide "maximum situativity and authenticity" (Brovelli et al., 2014, p. 555). We hope that future research will work to develop authentic video clips of CT embedded in classroom instruction. Accordingly, future work should use these clips to engage teachers in productive conversation on how they view role of CT within language arts, math, and science (Sherin, Linsenmeier, & van Es, 2009). Using classroom video as an assessment tool would allow researchers to characterize teacher thinking around CT integration. At the same time, such video-based discussions are even more difficult to systematically capture and code than were our vignette-based prompts. We argue that vignette-based prompts provide a productive middle ground: they "un-black-box" teacher thinking to an extent, while still maintaining replicable structure and forms of data collection that make traditional surveys a compelling research instrument. Continued work in developing and applying coding schemes to teachers' responses to vignette-based open-ended prompts is an important area for future development of such assessments.

Finally, these assessments are not yet capturing an important aspect of teacher learning: whether and how teachers actually translate CT integration into their instruction. Future research on this topic should include not only text or video vignettes, but also couple them with classroom observations and interviews to see how teachers take up computational thinking ideas in the context of their classrooms, and how these "seeds" of computational thinking might support future science and computer science learning. Such studies would truly "un-black-box" teacher thinking about integrating CT. Specifically, we need to study how teachers CT ideas play out in classroom lessons and how they could be used to facilitate their students' understanding of scientific concepts. The classroom observations would provide a lens into how teachers take up computational thinking ideas not as finished products in and of themselves, but as invitations to explore teaching and learning within specific subject area context (Segall, 2004). Additionally, the observations would help researchers examine what doors CT opens and what doors CT closes for teachers (and their students) when teaching (and learning) disciplinary ideas (Segall).

In addition, given that we also view computational thinking as a tool to help students see the relevance of computer science (even pursue computer science), it is important to examine how conceptualizing CT as a tool for science could help students prepare for computer science. For example, a study could investigate how students who come to look at data patterns and predict outcomes (aka IF (this pattern) – THEN (this outcome) – ELSE (that outcome)) in the context of science could apply conditional logic when learning to program. In summary, we need to continue a program of research that develops an empirical basis for developing an understanding of how teachers learn to integrate CT in meaningful ways into their classroom instruction.

Note

1. The number represents the responses coded under a theme and does not reflect the number of participants.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

This work was supported by the National Science Foundation under grant number 1543061

Notes on contributors

Dr. Aman Yadav is a Professor in Educational Psychology and Educational Technology Program and Director of Masters of Arts in Educational Technology at Michigan State University. His research focuses on educating preservice and inservice teachers to teach computer science and incorporate computational thinking in elementary and secondary classrooms.

Dr. Christina Krist is an Assistant Professor of Science Education at the University of Illinois at Urbana-Champaign. Her research focuses on understanding how to make science learning meaningful for students and teachers.

Dr. Jon Good is a research consultant at Oakland Schools in Waterford, Michigan. His research focuses on computer science education, computational thinking, and creativity.

Elisa Nadire Caeli is a doctoral student at the Danish School of Education, Aarhus University, and Department of School and Learning, University College Copenhagen. Her research focuses on children's development of computational thinking skills in elementary and secondary education to prepare for a 21st century democratic society.

ORCID

Aman Yadav http://orcid.org/0000-0003-4247-2033
Christina Krist http://orcid.org/0000-0002-9738-4308
Elisa Nadire Caeli http://orcid.org/0000-0002-1233-7589

References

Aarhus University (2017). Computational thinking and design to become a mandatory part of curriculum in Danish primary school. *Center for Computational Thinking and Design, Aarhus University*. Retrieved from http://cct.au.dk/currently/news/show/artikel/computational-thinking-and-design-to-become-a-mandatory-part-of-curriculum-in-danish-primary-school/

Barnhart, T., & van Es, E. (2015). Studying teacher noticing: Examining the relationship among pre-service science teachers' ability to attend, analyze and respond to student thinking. *Teaching and Teacher Education*, *45*, 83–93.

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.

Blum, L., & Cortina, T. J. (2007). CS4HS: An outreach program for high school CS teachers (pp. 19–23). In *Proceedings of the 38th SIGCSE technical symposium on Computer science education (SIGCSE '07)* (pp. 19–23). New York, MY: ACM. doi:10.1145/1227310.1227320

Bocconi, S., Chioccariello, A., Dettori, G., Ferrari, A., & Engelhardt, K. (2016). *Computational thinking in compulsory education*. Technical report, European Union Scientific and Technical Research Reports, 2016. EUR 28295 EN.

Bower, M., & Falkner, K. (2015). Computational thinking, the notional machine, pre-service teachers, and research opportunities. *Proceedings of the 17th Australasian Computing Education Conference*, ACE. Retrieved 2015, (January), 37–46 from http://crpit.com/confpapers/CRPITV160Bower.pdf

Brackmann, C. P., Román-González, M., Robles, G., Moreno-León, J., Casali, A., & Barone, D. (2017). Development of computational thinking skills through unplugged activities in primary school. In *Proceedings of the 12th Workshop in Primary and Secondary Computing Education* (pp. 65–72).

Brovelli, D., Bölsterli, K., Rehm, M., & Wilhelm, M. (2014). Using vignette testing to measure student science teachers' professional competencies. *American Journal of Educational Research*, *2*(7), 555–558.

Bundy, A. (2007). Computational thinking is pervasive. *Journal of Scientific and Practical Computing*, *1*(2), 67–69.

Calao, L. A., Moreno-León, J., Correa, H. E., & Robles, G. (2015). Developing mathematical thinking with scratch. In G. Conole, T. Klobučar, C. Rensing, J. Konert, and E. Lavoué (Eds.), *Design for teaching and learning in a networked world* (pp. 17–27). Cham: Springer.

Charmaz, K. (2006). *Constructing grounded theory*. London: Sage.

Corradini, I., Lodi, M., & Nardelli, E. (2017). Conceptions and misconceptions about computational thinking among Italian primary school teachers. *Proceedings of the 2017 ACM Conference on International Computing Education Research - ICER '17*, Tacoma, WA, USA (pp. 136–144).

Denning, P. J. (2017, June). Remaining trouble spots with computational thinking. *Communications of the ACM*, *60*(6), 33–39.

Denning, P. J., & Freeman, P. A. (2009). The professions of IT. Computing's paradigm. Viewpoints. *Communications of the ACM*, *52*(12), 28.

Dyer, E. B. (2016). *Learning through teaching: An exploration of teachers' use of everyday classroom experiences as feedback to develop responsive teaching in mathematics* (Doctoral Dissertation). Northwestern University.

Dyer, E. B., & Kaliski, P. (2016). Secondary mathematics teachers' use of causal reasoning about classroom experiences to change teaching practice. *Paper presented at the Annual Meeting of the American Educational Research Association*, Washington, D.C.

EMU (2017). *Teknologiforståelse valgfag (forsøg) – Fælles Mål og læseplan*. Retrieved from https://www.emu.dk/modul/teknologiforst%C3%A5else-valgfag-fors%C3%B8g-%E2%80%93-f%C3%A6lles-m%C3%A5l-og-l%C3%A6seplan

Good, J., Yadav, A., & Mishra, P. (2017). Computational thinking in computer science classrooms: Viewpoints from. In C. S. Educators, P. Resta, & S. Smith (Eds.), *Proceedings of society for information technology & teacher education international conference* 51–59. Austin, TX, United States: Association for the Advancement of Computing in Education (AACE). Retrieved October 13, 2017 from https://www.learntechlib.org/p/177274/

Grover, S., & Pea, R. (2013). Computational thinking in K–12: A review of the state of the field. *Educational Researcher*, *42*(1), 38–43.

Horn, I. S. (2005). Learning on the job: A situated account of teacher learning in high school mathematics departments. *Cognition and Instruction*, *23*(2), 207–236.

Knuth, D. E. (1974). Computer programming as an art. *CACM*, December. http://www.paulgraham.com/knuth.html

Korthagen, F. A. (2010). Situated learning theory and the pedagogy of teacher education: Towards an integrative view of teacher behavior and teacher learning. *Teaching and Teacher Education*, *26*(1), 98–106.

Krist, C., Elby, A., Good, J., Gupta, A., Sohr, E. R., & Yadav, A. (2017, April). Integrating computational thinking strategies that support science inquiry: A case study from a summer PD. Paper presented at the Annual Meeting of American Educational Research Association, San Antonio, TX.

Malmberg, A. C. (1970). Datalogi i skolen: Læreruddannelsen – En flaskehals. *Undervisningsministeriets tidsskrift* 272–276.

Miles, M. B., & Huberman, M. (1994). *Qualitative data analysis: An expanded sourcebook*. Newbury Park, CA: Sage.

Mouza, C., Yang, H., Pan, Y., Ozden, S. Y., & Pollock, L. (2017). Resetting educational technology coursework for pre-service teachers: A computational thinking approach to the development of technological pedagogical content knowledge (TPACK). *Australasian Journal of Educational Technology*, *33*(3), 61–76.

NGSS Lead States. (Ed.) (2013). *Next generation science standards: For states, by states*. Washington, D.C: National Academies Press.

Sands, P., Yadav, A., & Good, J. (2018). Computational thinking in K-12: In-service teacher perceptions of computational thinking. In M. S. Khine (Ed.), *Computational thinking in the STEM disciplines* (pp. 151–164). Cham: Springer.

Santagata, R., & Yeh, C. (2016). The role of perception, interpretation, and decision making in the development of beginning teachers' competence. *ZDM*, *1*. doi:10.1007/s11858-015-0737-9

Segall, A. (2004). Revisiting pedagogical content knowledge: The pedagogy of content/the content of pedagogy. *Teaching and Teacher Education*, *20*, 489–504.

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

Sherin, M., Jacobs, V., & Philipp, R. (Eds.). (2011). *Mathematics teacher noticing: Seeing through teachers' eyes*. New York: Taylor and Francis.

Sherin, M., & van Es, E. (2005). Using video to support teachers' ability to notice classroom interactions. *Journal of Technology and Teacher Education*, *13*(3), 475–491.

Sherin, M. G., Linsenmeier, K. A., & van Es, E. A. (2009). Selecting video clips to promote mathematics teachers' discussion of student thinking. *Journal of Teacher Education*, *60*, 213–230.

Tedre, M., & Denning, P. J. (2016) The long quest for computational thinking. *Proceedings of the 16th Koli Calling Conference on Computing Education Research*, November 24–27, 2016, Koli, Finland (pp. 120–129).

Thornberg, R. (2012). Informed grounded theory. *Scandinavian Journal of Educational Research*, *56*(3), 243–259.

Van Es, E. A., & Sherin, M. G. (2002). Learning to notice: Scaffolding new teachers' interpretations of classroom interactions. *Journal of Technology and Teacher Education*, *10*(4), 571–596.

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

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, *49*(3), 33–35.

Yadav, A., Good, J., Voogt, J., & Fisser, P. (2017). Computational thinking as an emerging competence domain. In M. Mulder (Ed.), *Competence-based vocational and professional education* (pp. 1051–1067). Switzerland: Springer International Publishing. Retrieved from. http://link.springer.com/chapter/10.1007/978-3-319-41713-4_49

Yadav, A., Berges, M., Sands, P., & Good, J. (2016). Measuring computer science pedagogical content knowledge: An exploratory analysis of teaching vignettes to measure teacher knowledge. *Paper presented at Wipsce*, Munster, Germany.

Yadav, A., Hong, H., & Stephenson, C. (2016). Computational thinking for all: Pedagogical approaches to embedding a 21st century problem solving in K-12 classrooms. *TechTrends*, 60, 565–568. doi:10.1007/s11528-016-0087-7

Yadav, A., Mayfield, C., Zhou, N., Hambrusch, S., & Korb, J. T. (2014). Computational thinking in elementary and secondary teacher education. *ACM Transactions on Computing Education*, *14*(1), 1–16.

Yadav, A., Sands, P., Good, J., & Lishinski, A. (2018). Computer science and computational thinking in the curriculum: Research and practice. In J. Voogt, G. Knezek, R. Christensen, & K. -W. Lai (Eds.). *Handbook of Information Technology in Primary and Secondary Education*. Cham: Springer International Handbooks of Education. doi:10.1007/978-3-319-53803-7_6-1

Yadav, A., Stephenson, C., & Hong, H. (2017). Computational thinking for teacher education: A framework for developing teacher knowledge. *Communications of the ACM*, *60*(4), 55–62.

Yadav, A., Zhou, N., Mayfield, C., Hambrusch, S., & Korb, J. T. (2011). Introducing computational thinking in education courses. In *Proceedings of the 42nd ACM technical symposium on Computer science education* (pp. 465–470). ACM. Retrieved from http://dl.acm.org/citation.cfm?id=1953297

Appendix A. Unplugged Computational Thinking Activities During Teacher PD Sessions

Goals and rationale. To introduce participants to computational thinking, we designed two activities that aimed to first have them experience and use unplugged computational thinking in an everyday context, then reflect on those experiences and attach more formal names to the kinds of thinking that they described using during the activity. We aimed to design tasks that were relatively open-ended in that multiple solutions were possible; and that were "everyday" activities such that participants would be able to draw on their own experiences as examples of nascent computational thinking.

Computational thinking activity 1: Getting to the National Mall. In this activity, which occurred during Week 1 of the summer workshop in Summer 2016, facilitators asked participants to write out directions for the *best way* to get to the National Mall from their current location on a university campus. The definition of "best" was intentionally left open for interpretation. Participants could use any resources they needed, including their phones or computers. Facilitators also asked participants to note the kinds of information they searched for, if any, and any decisions they consciously made as they were creating their directions. After working individually, each participant shared out to the group. The facilitators led a reflective conversation about how the sets of directions were similar and different and why, and also what kinds of decisions and information they used and why. These reflective conversations included the following questions:

- What pieces of information did you want to know?
- How did you each define "best"?
- What were some of the decisions you made while writing directions? What was the process you went through in making these directions?
- Did you go with your first design? Did you change direction because of new information, or did the process reveal new issues?
- What would you change now after discussion?
- How did you choose to represent the directions?

Facilitators then asked participants to combine their directions into one set that would take into account all the complexity they had each designed for. This task was again followed by a reflection on the process similar to the one described above, as well as an introduction of some formal CT terminology to label some of the practices and skills they had used for the activity.

From this activity, participants described (and facilitators re-named) six different computational thinking concepts and strategies: making if/then decisions and/or including them as part of the structure of their instructions; seeking out relevant data or information to build the instructions; tradeoffs between simplicity, efficiency, and detail; debugging and iteration while developing instructions; chunking, or decomposing the directions such that they could ignore certain portions while working on others; and running algorithms.

Computational thinking activity 2: Expedia Sort. The second computational thinking activity occurred during Week 2, Day 3 of the workshop during Summer 2016. Facilitators asked participants to "be Expedia" for a user who was searching for a plane ticket: specifically, to decide what kinds of rules they would use to sort and display the search results. After doing so, facilitators led a reflective discussion similar to the one described above. As part of this discussion, facilitators identified new computational thinking concepts or strategies that the participants were using or describing. These new ideas were added to the running list of computational thinking strategies.

The two groups participating in this activity took it up very differently from each other. One group was focused on *how* the computer sorted, as in the mechanics of numerical sorting, and writing out a set of instructions that said first look at the thousands place, then the hundreds place, etc. The other group was focused on the hierarchy of making decisions for presenting the information to customers. For example, is it better to list a cheaper flight with three layovers before a slightly more expensive but direct flight? This question eventually spread to both groups, and the discussion transitioned to thinking about optimizing the search-and-sort process: what is the fewest number of questions you could ask a customer and still provide them with good-enough options?