Infusing Computational Thinking Instruction into Elementary Mathematics and Science: Patterns of Teacher Implementation

Kathryn M. Rich
College of Education, Michigan State University, East Lansing, MI, USA
richkat3@msu.edu

Aman Yadav
College of Education, Michigan State University, East Lansing, MI, USA
ayadav@msu.edu

Abstract: In this brief paper, we will share preliminary results of a study of how elementary-school teachers take up computational thinking (CT) ideas and incorporate them into their mathematics and science teaching. We describe the teachers' school contexts, the professional development experiences in which they engaged, and our preliminary analyses of how they used computational thinking within their enacted lessons. In brief, the seven teachers in this study exhibited three patterns of implementation: (1) using computational thinking to guide their own planning and thinking; (2) using computational thinking to structure their lessons; and (3) presenting computational thinking concepts to students as general problem solving strategies.

Context and Motivation

In recent years, there have been increasing calls, in the United States and elsewhere, to bring quality computer science instruction to all students in K-12 (Grover & Pea, 2013; Yadav, Hong, Stephenson, 2016). Given the role computing plays in today's digitized world, some have argued it can be seen as a new form of literacy appropriate for all (Vee, 2017) and are in favor of exposing students to *computational thinking,* which is loosely defined as the mental tools, such as abstraction and decomposition, used within computer science to solve problems (Wing, 2006).

Motivated, in part, by the need to fit computational thinking instruction into an already full school day, several recent efforts at the elementary school level have aimed to integrate computational thinking ideas into core subjects such as mathematics (e.g., Israel, Pearson, Tapia, Wherfel, & Reese, 2015; Rich, Strickland, Binkowski, Moran & Franklin, 2017). Much of this work focuses on exposing students to computational thinking ideas through programming. Less attention has been given to the prospect of introducing computational thinking ideas in unplugged contexts. Our study focuses on introducing teachers, and then students, to computational thinking ideas embedded within their existing mathematics and science practices, thereby giving teachers a familiar, unplugged context in which to explore and build their own understanding of these ideas. This paper reports preliminary results on how teachers took up computational thinking and integrated it in their classroom.

Participants

Participants were seven elementary school teachers from a large intermediate school district surrounding an urban area in the Midwest United States. They were currently teaching grade 3, 4, or 5 and varied in their teaching experience from 3 to more than 20 years. Their school demographics varied, but the schools were chosen, in part, based on high percentages of minority students, students from low-income families, or English-language learners.

Professional Development

Teachers participated in three professional development workshops as a part of an NSF-funded project (CT4EDU: http://ct4edu.org/home/) prior to implementing their CT lessons in their classrooms. In Spring of 2018, they received a preliminary introduction to computational thinking and discussed the ways in which CT ideas might apply to their mathematics and science teaching. In early Summer of 2018, they were formally introduced to four computational thinking ideas on which the remainder of the project would focus: *abstraction* to identify important information within complex contexts, *decomposition* of problems into easier-to-manage parts, recognizing and applying *patterns* to connect problems to each other, and *debugging,* or finding and fixing errors.

In late Summer of 2018, teachers came to the third workshop with examples of their existing mathematics or science lessons (often from a district-provided textbook or other resource) that they felt were good candidates for CT integration. Teachers used a *CT Lesson Screener* tool to identify elements of CT already present in the lesson. They then used a *CT Lesson Enhancer* tool to make specific plans for implementing the lessons in ways that would make existing CT ideas more explicit or embed new opportunities for CT (See Yadav, Larimore, Rich, & Schwarz, 2019 for a description of these tools).

Implementation and Data Collection

The teachers each implemented at least one of their planned CT lessons during the first half of the 2018-2019 school year. We used tablets mounted on *Swivl* devices to record video of each lesson implementation. Teachers wore microphones that were synced to the *Swivl* devices, so that the base rotated to follow the teachers as they moved around their classrooms. The first author was present as an observer for most of the implementations; the second author was also present for one lesson. Observers took field notes both during and immediately after implementation. Reflection on these observation experiences began discussions that raised the idea of examining patterns of implementation After discussing various potential patterns of implementation, we engaged in preliminary analysis of each teacher's lesson planning documents, discussions during professional development sessions, and classroom video. We triangulated information from these data sources to define and describe the three patterns of implementation described below.

Preliminary Results: Patterns of Implementation

The seven teachers fell into three categories. Specifically, they: (1) used CT to guide their planning, (2) used CT to structure their lessons, or (3) presented CT as general problem-solving strategies. In all the examples presented below, we use psuedonyms to refer to our partner teachers.

Using CT to Guide Teacher Planning

Two fifth-grade teachers primarily used computational thinking concepts to guide their own thinking and lesson planning, but rarely made the CT ideas explicit to students. For example, Ms. Anderson's lesson plan focused on abstraction through discussion of various representations of fractions. The mathematical goal of the lesson was for students to fluently convert among improper fractions and mixed numbers. In her *CT Lesson Screener,* Anderson noted that students would be engaging in abstraction during the lesson when they "Look at visual representations of mixed numbers, identifying the whole and the extra". This activity is explicitly noted on her *CT Lesson Planner* tool as follows: “Put visual on the board, ask students, What do you see? How many slices are shaded in total? What is the whole? Is this one whole, or two?”

During implementation, Anderson asked students her planned question of “What do you see?” – sometimes phrased as “How much do you see?” – about two different representations of fractions: a visual model of 5/4 (two squares, each divided into fourths, with one entirely shaded and one with one fourth shaded), and a sum of unit fractions for 5/3 (1/3 + 1/3 + 1/3 + 1/3 + 1/3). During the discussion, she encouraged students who shared answers to explain where they saw the 1 or the whole. For example, when a student said the sum of unit fractions was equal to 1 2/3, Anderson asked, “Can you explain? Because I don’t see a 1 anywhere up there, and I think some other friends are like, where is he getting 1 from or where is 2/3 from?” The student came to the board and drew a ring around 1/3 + 1/3 + 1/3, explaining that this was the same as 3/3, or 1. The word *abstraction,* however, was not used by the teacher or the student.

In short, we know from Anderson’s planning document that she saw the work of identifying the whole in various representations of fractions as a kind of abstraction. Still, she chose not to discuss these ideas in terms of abstraction with her students. Her plans and implementation reflected her own thinking about abstraction, but she did not make that thinking transparent to students. The second teacher who fell into this category, Ms. Alberts, similarly used abstraction to guide her thinking, but did not explicitly discuss abstraction with students.

Using CT to Structure Lessons

Two other teachers, one teaching third grade and one teaching fourth grade, chose a particular CT concept and structured a lesson around it. For example, Ms. Baldwin, the third grade teacher, organized a science lesson on force and motion around the CT idea of debugging. Students in her classroom constructed mini “rockets,” called *hopper poppers,* that launched from their tables powered by rubber bands. Her *CT Lesson Planner* shows that she planned to build in an opportunity for students to engage in debugging: “Once the students have finished testing their baseline hopper popper, the teacher will ask the students to share their results. The students will then be given an opportunity to debug their hopper popper to get a better result (higher jump).” She also planned to ask students the following question about debugging: “How are you modifying your hopper popper to reach a higher height?”

During implementation, Baldwin set up the activity of testing and modifying the hopper poppers as follows: “You’re going to have a chance to test your hopper popper. The one you already made. And then you’re going to have a chance to fix your hopper popper. OK, looking up at my CT wall, after we test our hopper popper and we get a chance to fix it, raise your hand if you see what CT skill we’re going to be using.” The CT wall refers to a section of the classroom wall showing posters of the four CT ideas emphasized in the project: abstraction, decomposition, patterns, and debugging. A student identified debugging as the relevant skill for the hopper popper testing.

In contrast to Anderson and Alberts, Baldwin not only used a CT idea (debugging) to guide her own thinking in lesson planning, but also engaged students in thinking about and using debugging strategies. She framed debugging as a central goal of the lesson, structuring the lesson around it and using the word *debugging* with students. The other teacher who fell into this category, Ms. Burns, structured a mathematics lesson around the CT idea of abstraction. She engaged them in discussing what information about exact sums might be lost when they used rounding strategies to estimate sums.

Presenting CT Ideas as General Problem-Solving Strategies

The remaining three teachers, one teaching fourth grade and the other two teaching fifth grade, did not make a particular CT concept the focus on their lessons, but they did make the CT ideas explicit to students in a different way. These teachers introduced particular tasks to students, and then discussed the CT ideas as strategies that could be used to solve the problems. For example, Mr. Carter’s *CT Lesson Planner* document showed that he intended to have students solve a problem involving whole-number place value any way they wanted, then follow up by looking for the four CT ideas in students' strategies. His planned questions and prompts around each CT idea were as follows (emphasis added): "*Decompose* your group's number and write just your digit's value on your board, and say it using name form. Use *abstraction* to have each person share their expanded form value - don't start until your neighbor says 'plus.' What *patterns* do you see moving from one place to the next? Why? Don't forget to *debug*. How can you check your work to see if your answer(s) work?"

Carter used a different mathematics topic, comparing decimals, as the focus of his CT-enhanced lesson, so he did not use his planned questions and prompts exactly as written. However, his implementation also reflected a general approach of presenting the CT ideas as general problem-solving strategies. He started his mathematics lesson by reviewing the four CT ideas (decomposition, abstraction, debugging, and patterns), asking students to state in their own words their understanding of each idea. Then he had them solve a problem involving ordering decimals any way they wanted. As the lesson progressed, Carter referenced the CT ideas at moments when he was able to connect them to what students were doing. For example, as students worked, he suggested that they compare with their partners or groups and "do some debugging." When the textbook problem students were reading mentioned patterns, he said, "This problem, we're going to be looking for patterns all over it." He then followed up by highlighting the book's mention of breaking a problem into simpler parts, asking, "Which one of our computational thinking strategies is that?"

Thus, in contrast to Anderson and Alberts (but like Baldwin and Burns), Carter explicitly used computational thinking language in his classroom. In contrast to Baldwin and Burns, he did not use the CT ideas to structure his lesson, but rather brought them up in opportunistic ways as they became relevant to the discussion. Two other teachers followed a similar implementation pattern. A fourth grade teacher, Ms. Collins, challenged students to generate a rule they could use to round numbers by looking for patterns in their answers, noting that looking for patterns is a strategy that computer scientists use. When students had discovered the rounding rule (round down if the digit to the left is 4 or less, otherwise round up), Collins also pointed out that the rule was an abstraction of the number-line strategy. A fifth grade teacher, Mr. Curtis, encouraged students to develop theories for why the sun rises and sets by using abstraction to focus in on observation details that might matter, and to debug their thinking when their theories did not match up with their observations.

Implications

This study contributes to a growing body of reasearch exploring how elementary school teachers might understand and apply computational thinking ideas in their mathematics and science teaching (e.g., Duncan, Bell, & Atlas, 2017; Israel et al., 2015). Specifically, we shared ways that elementary teachers incorporated CT ideas into their mathematics and science teaching in unplugged contexts, tracing connections between their plans and their implementations. These preliminary results suggest three ways that elementary school teachers might use CT in their classrooms. Our findings could inform future research to examine how these patterns of implementation might relate to teachers' and students' understanding of CT concepts. We present the following reflections and question to guide that work.

The first pattern of implementation (*using CT to guide teacher planning*) does not explicitly expose students to CT, but does engage them in kinds of thinking that relate to computer science. This raises the question: at what point is it necessary or useful to make CT explicit to students?

The second pattern of implementation (*using CT to structure lessons*) makes CT ideas a clear focus in students' experiences, but the CT ideas are situated in mathematics and science. What impact might this have on students' understanding of CT when applied to computer science? Does early exposure in a familiar context support their later use of CT in more unfamiliar contexts?

The third pattern of implementation (*presenting CT ideas as general problem-solving strategies*) also situates CT in mathematics and science, but does so in a way that makes fewer or weaker connections to specific mathematics and science concepts. Does such an implementation pattern support transfer of CT to computer science?

Acknowledgement: This work is supported by the National Science Foundation under grant number 1738677. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

Duncan, C., Bell, T., & Atlas, J. (2017). What do the teachers think ? Introducing computational thinking in the primary school curriculum. In *Proceedings of the Nineteenth Australasian Computing Education Conference* (pp. 65–74).

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

Israel, M., Pearson, J. N., Tapia, T., Wherfel, Q. M., & Reese, G. (2015). Supporting all learners in school-wide computational thinking: A cross-case qualitative analysis. *Computers and Education*, *82*, 263–279.

Rich, K. M., Strickland, C., Binkowski, T. A., Moran, C., & Franklin, D. (2017, August). K-8 learning trajectories derived from research literature: Sequence, repetition, conditionals. In *Proceedings of the 2017 ACM Conference on International Computing Education Research* (pp. 182-190). ACM.

Vee, A. (2017). *Coding literacy: How computer programming is changing writing.* The MIT Press: Cambridge, MA.

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

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.

Yadav, A., Larimore, R., Rich, K., & Schwarz, C. (2019). Integrating computational thinking in elementary classrooms: Introducing a toolkit to support teachers. Paper presented at the annual meeting of the Society for Information Technology in Teacher Education (SITE), Las Vegas, NV.