CJLT RCAT
Canadian Journal of Learning and Technology
La Revue canadienne de l'apprentissage et de la technologie

Volume 45(3) Fall/Automne 2019

Computational Thinking in Classrooms: A Study of a PD for STEM Teachers in High-Needs Schools

Pensée informatique dans les salles de classe: Une étude d'un PD pour les enseignants STEM dans les écoles à besoins élevés

Qing Li, Towson University
Laila Richman, Towson University
Sarah Haines, Towson University
Scot McNary, Towson University

Abstract

This study explores the influence of a professional development (PD) model aiming to build teacher capacities for K-12 schools. It examines the impact of this PD on teachers' learning of content and pedagogical knowledge related to computational thinking (CT). It also investigates the lessons learned during the implementation process.

This mixed-methods study examined 25 teachers who participated in the PD. The pre- and post-test analysis showed positive outcomes of this PD in helping teachers learn CT skills. The thematic analysis of the qualitative data identified themes to reveal how teachers integrate CT into their lesson plans and classroom practices, as well as lessons learned. Learner-centered multidisciplinary approaches, differentiated learning, and unplugged activities were three main themes identified in teacher created lesson plans. Specific recommendations based on the lessons learned include offering PD over an extended period of time, blended learning, and promoting collaboration.

Résumé

Les concepteurs pédagogiques sont dans une position unique pour exercer un leadership et un soutien pour l'avancement des nouvelles technologies et pratiques. Pourtant, une recherche documentaire révèle un manque de recherche sur les rôles actuels et potentiels des concepteurs pédagogiques en ce qui concerne l'intégration et la promotion de pratiques éducatives ouvertes dans leurs établissements d'enseignement supérieur. Dans le contexte des nouvelles pratiques éducatives ouvertes, une enquête et des entretiens ont été menés auprès de concepteurs pédagogiques pour établir, à partir de leur expérience et de leurs pratiques, leurs rôles et leur potentiel pour supporter des pratiques éducatives ouvertes (OEP), y compris les ressources éducatives libres (OER). Parmi les résultats de l'analyse, il a été constaté que, bien que les concepteurs pédagogiques soient fortement conscients et désireux de défendre les OEP dans leurs établissements, leur capacité à aller de l'avant était limitée par des obstacles perçus tels que le manque de mandats pertinents et la non reconnaissance de la charge de travail professionnelle additionnelle, le développement de politiques et de financement, la sensibilisation et le soutien au leadership. De plus, des écarts ont été identifiés entre ce qu'ils apprécient le plus au sujet des OEP, comme la mise en œuvre de pédagogies innovantes, et ce qu'ils pourraient réellement initier et supporter en pratique (adopter et soutenir les OER). Ils ont souligné le manque d'opportunités formelles d'apprentissage au regard des OEP et ont révélé que leurs principales sources d'apprentissage et de soutien étaient de nature informelle, acquises par le biais de leurs réseaux et collaborations avec des pairs.

Introduction

Computational thinking (CT) has gained growing attention in recent years from various groups, acknowledging that it has become a new “basic skill” that every K-12 student needs to master. A significant challenge for CT education is the lack of time for stand-alone CT courses (Williams, 2017). Integrating CT into existing curriculum in PreK-12 classrooms to help students develop such basic skills has a great potential to address this challenge (Williams, 2017; Yadav, Stephenson, & Hong, 2017). However, basic development of curricula and adequate preparation of teachers for CT skills are still works in progress (Grover & Pea, 2013). How CT can be best integrated into K-12 subjects other than computer science is still unknown (Voogt, Fisser, Good, Mishra, & Yadav, 2015). Well-trained teachers with the needed skills who can effectively integrate CT in classrooms are scarce and face personal and systemic obstacles (Bower, Wood, Lai, Howe, Lister, Mason, Highfield, & Veal, 2017; Israel, Pearson, Tapia, Wherfel, & Reese, 2015). Preparing all teachers with adequate content and pedagogical knowledge related to CT integration is a logical and necessary first step in order to infuse CT into K-12 education.

One approach is to provide professional development opportunities to introduce CT to practicing teachers. In this study, we explore the influence of a professional development model that aims to build teacher capacities for K-12 schools with a special focus on high need schools.

Related Literature

Defining CT

In 2006, Jeannette Wing coined the term computational thinking which started a profound discussion related to the role of CT across disciplines (Barr & Stephenson, 2011). To date, various definitions exist. Some closely relate to computer science while others focus on operational definitions for K-12 education. For example, Wing defined CT as “the thought processes involved in formulating a problem and expressing its solution(s) in such a way that a computer-human or machine-can effectively carry out” (2014, p. 6).

Others such as the International Society for Technology in Education (ISTE) and the Computer Science Teachers Association (CSTA) operationally define CT as: a problem-solving process that includes (but is not limited to) the following characteristics:

- Formulating problems in a way that enables us to use a computer and other tools to help solve them
- Logically organizing and analyzing data
- Representing data through abstractions such as models and simulations
- Automating solutions through algorithmic thinking (a series of ordered steps)
- Identifying, analyzing, and implementing possible solutions with the goal of achieving the most efficient and effective combination of steps and resources
- Generalizing and transferring this problem-solving process to a wide variety of problems (ISTE, 2011, para. 2)

In addition to the important cognitive aspects, affective components such as confidence and persistence are equally critical when considering CT.

It is important to realize that in education, two distinct perspectives exist in the literature related to CT. In his review paper, Nardelli (2019) offers a comprehensive articulation of these two perspectives. He states that the first perspective considers CT as a new subject differing from CS, while the second angle views CT as thinking habits acquired through computer science (CS) learning. Through the examination of CT's historical evolvement, he argues that CT is part of CS focusing only on the "scientific and cultural aspects of computing… not dealing with system and tools, but with principles and methods" (p. 32). Regardless of the viewpoints, most, if not all, scholars agree that CT needs to be integrated into subjects beyond CS.

CT and Teacher Training

CT is not a new concept. In fact, the idea was introduced into the academic discourse as early as 1962 but under different names and definitions (Czerkawski & Lyman, 2015). Seymour Papert's pioneer work with LOGO in the 1980s generated the first wave of great interest in CT in K-12 classrooms (Grover & Pea, 2013). Recent work considers CT as a basic skill that needs to be obtained by everyone rather than just experts in computer related fields (Grover & Pea, 2013; Oluk & Korkmaz, 2016). The growing interest in CT and CS in K-12 education resulted in an increasing number of research studies, yet major efforts were focused on K-12 student learning with little attention paid to teacher training (Voogt et al., 2015).

Though limited, some efforts were made to help teachers develop competencies to integrate CT in their classrooms. Yadav and colleagues (Yadav, Mayfield, Zhou, Hambrusch, & Korb, 2014) conducted a study examining the impact of a one-week PD on preservice teachers. The participants, who were enrolled in a mandatory introductory educational psychology course, were divided into a treatment (complete a CT module) and a control (no CT module) group. The analysis of the data showed that the PD enhanced teachers' understanding of CT; most participants in the control group equated CT to technology integration, while teachers in the treatment group viewed CT as problem solving via algorithms/heuristics. Compared to the control group, teachers in the treatment group were better able to discuss how CT could be integrated in classrooms.

An Australian study (Bower & Falkner, 2015) examined preservice teachers' perceptions of CT learning and teaching through a survey of 44 students (33 females, 11 males) enrolled in a 300-level education course. They found that for most preservice teachers, CT was a confusing concept that was equated to general technology use (e.g., searching the Internet). For many, CT integration into teaching meant to use technology in classrooms. Participants' confidence level in teaching CT varied, with a majority of them lacking confidence while several were overconfident and demonstrated misconceptions. The preservice teachers also welcomed opportunities to enhance their content, pedagogical, and technological knowledge related to CT teaching.

A Canadian study (Gadanidis, Cendros, Floyd, & Namukasa, 2017) examined 143 preservice elementary teachers' experience of learning CT in a math education course. The blended nine-week course focused on helping teachers "develop conceptual understanding of mathematic and mathematics teaching with CT" (p. 459). Employing a case study approach, the paper analyzed teacher online discussions and reflection assignments. They found that teachers in this short 18-hour course developed some new ideas related to CT. The course also helped reduce teacher apprehension towards CT in mathematics teaching and learning.

Additionally, Yadav et al., (2017), reviewed existing work and discussed why CT is important for teacher training and how CT should be embedded in teacher education. Focusing on preservice education, they described specific steps and mechanisms to expose CT to preservice teachers.

In sum, research into how to prepare teachers for the integration of CT into curriculum is still lacking (Grover & Pea, 2013; Yadav et al., 2017). This study, therefore, focuses on teacher experience of a PD project aimed to help them gain content and pedagogical knowledge of CT integration in STEM education.

Theoretical Framework

The theoretical framework of this paper is "enactivism" applied in educational technology, which is rooted in biology (Varela, Thompson, & Rosch, 1991) and phenomenology (Merleau-Ponty, 1964). Fundamentally different from other prevailing theories like constructivism or behaviorism, enactivism rejects dualism and focuses on the importance of embodiment and action to cognition. In stressing embodied action, it finds a middle way between two extreme views about reality: the objective view assumes that reality exists independent of our experience versus the subjective perspective in which reality is independent of the surrounding world.

Compatible with elements of Piaget's and Vygotsky's psychology as well as experientialism of Lakoff (Reid, 1995), enactivism, is based on two important premises, (a) cognition and environment are inseparable, and (b) "systems" enact with each other from which they "learn." It argues that "the world is inseparable from the subject, but from a subject which is nothing but a project of the world, and the subject is inseparable from the world, but from a world which the subject itself projects" (Varela et al., 1991, p.7). Cognition is therefore, a human, social, and biological phenomenon. Learning occurs through the learners' acts and is acted upon by the learning world.

The first central idea of enactivism is its focus on doing. A well-known slogan of enactivism is "all *doing* is knowing and all-knowing is *doing*" (Varela et al., 1991). However, it is important to note that enactivism is not a new label for "learning by doing." While doing plays a significant role in enactivism, earlier work (e.g. Fenwick, 2000; Li, 2014) has convincingly argued that enactivism is a fundamentally different viewpoint from "learning by doing." Considering limited spaces available, this is not elaborated here. Technology-supported active learning grounded in action offers ideal ways to promote such embodied cognition. The core focus of gaming and unplugged activities on *doing* provides a condition that is essential to craft an enactivist learning world (Li, 2014).

A second essential concept of enactivism is the emphasis on learner coauthoring rather than simply consuming knowledge. Learning, according to enactivism, occurs through feedback within the system (Fenwick, 2000). As such, cognition is a complex process of people interacting and affecting each other and their environments (Davis, Sumara, & Luce-Kapler, 2008). Leading-edge technology-mediated learning environments, like massive open online courses (MOOCs), provide a perfect platform to foster enactivist learning through collaborative knowledge building.

Professional Development

A comprehensive review of existing literature and an evaluation of a national survey (Desimone & Garet, 2015) led to the identification of 5 key supporting strategies of effective PD: (a) content focus, (b) active rather than passive learning, (c) alignment with teachers' beliefs, needs, and background, (d) long term (20 or more hours during the school year) and (e) teacher collaboration. In addition to these 5 key components, research (e.g. Blanchard, LePrevost, Tolin, & Gutierrez, 2016; Mouza, 2009) shows that technology has the potential to foster sustained transformation of teacher practices. For example, blended learning, which integrates online and face-to-face experiences, affords flexibility to engage teachers in just-in-time learning, reduces teacher anxiety, and fosters reflective practice (Kell, Rupley, Nichols, Nichols, Paige, & Rasinski, 2016). Further, using technology like games and simulations provides advantages because they increase learner engagement, and enable learning in meaningful contexts (Li, 2014).

Our PD model, therefore, builds on the research evidence and is guided by enactivism through the supporting strategies as defined in Table 1. The detailed description of the PD program is presented in the Methods section.

Table 1

PD Program

Key components	**Supporting strategies**	**Enactivist aspects**
Content focus	• Activities focus on CT in the context of STEM subjects that teachers are currently teaching. • Emphasis on doing by letting learners (i.e. teachers) construct artifacts as ways to represent connections between ideas.	• Coauthoring
Active learning	• Teachers immerse in hands-on activities, digital games and simulations with feedback, apply, implement, design and build learning materials rather than listening passive lectures. • Emphasis on learner co-constructing knowledge. Learner works are shared frequently for knowledge sharing.	• Doing • Coauthoring
Coherence	• Content is aligned with teacher needs, beliefs, and their curriculum (e.g. NGSS, MCCRS). • Online tools of CT skills.	• Coauthoring • Co-emerging

Sustained duration	• Contains at least 85 contacting hours throughout the school year.	• Doing
Teacher collaboration	• Teachers of the same grades/subjects /schools work collaboratively to build learning communities. • MOOC supported community building beyond schools.	• Coauthoring • Co-emerging
Technology-enhanced	• Blended online and face-to-face experiences using the MOOC provider EdX. • Digital games and simulations.	• Co-emerging • Doing • Coauthoring

Research Questions

This study aims to understand the effect of a blended PD on teachers' understanding and ways of integrating CT, as well as the lessons learned during the implementation process. The effect of the PD on imparting CT to K-12 learners, however, is beyond the scope of this study. Specifically, it is guided by these questions:

1. Does the blended PD influence teachers' understanding of CT?
2. How do teachers integrate CT as reflected in their lesson plans?
3. In what ways do teachers integrate CT into their own classroom practice?
4. What are lessons learned?

Methods

Adopting a mixed-methods approach, this study focused on examining the influence of blended PD on teachers. The data presented for this study was collected as part of a broader, multi-year project aimed at enhancing K-12 STEM teachers' understanding of content and pedagogical knowledge necessary to teaching CT in STEM areas. Institutional Review Board (IRB) approval was obtained before the data were collected and analyzed. All the teachers were teaching in high-need schools with Free and Reduced-Price Meal Program rates much higher than the state average.

The Context

A description of the PD is needed first to contextualize the study. The professional development program employed a blended learning approach, mixing online and face-to-face meetings.

1. The PD started with a three-hour face-to-face meeting where participants were introduced to the project leaders, mentors/coaches, technical supporters, and each other. The online course platform was also briefly introduced. Participants took various pre-surveys during this session.
2. For the next five months, participants worked on the online modules. A total of four online modules, comprising approximately 45 hours of work, were completed. These online modules focused on important CT concepts like modeling and included videos, text information, games, and hands-on activities to help familiarize teachers with the content. Various quizzes were also incorporated in the modules, providing teachers immediate feedback about their learning. The online component also offered opportunities for teachers to interact with each other and specialists in various areas, including experts in CT, math

and science educators, and technical supporters. During the online component, two optional face-to-face tutorial sessions were conducted to help teachers who might have technical difficulties.

3. A five-day summer institute (25 hours of face to face meetings) was conducted, aiming to help teachers understand how to integrate CT into their classroom practice. Teachers created lesson plans integrating CT. Post-surveys were also completed during this time.
4. The next six months were dedicated to a fifth online module. The teachers were also asked to further develop their CT integrated lessons and start implementing the lessons.
5. At the end of the sixth month, two colloquiums were held, allowing teachers to showcase their implementation work. Focus group interviews of the teachers were also conducted.

The blended PD approach was intentionally designed to provide teachers adequate time to engage with the content as their levels of CT and CS proficiencies varied. It is important to note, however, that the coaches maintained regular contact with the participating teachers as they completed the online portions and a few face-to-face support sessions were also offered for any teachers who needed additional support.

The online course, using EdX (a MOOC environment), started with an introduction session followed by five modules, with the first four modules focused on modeling using Starlogo Nova and last model took a game-based approach using Scratch. Both Starlogo Nova and Scratch are agent-based game and simulation programming environments. Below is a description of these modules:

1. The course introduction had a welcome page, brief overview of each module and material covered, expectations, the operational definition of CT by ISTE, and material to get teachers started (e.g., how to navigate the course, how to use the software StarLogo Nova).
2. The first module was an introduction of computer modeling focusing on science. Teachers first watched a video of real people enacting a simulation, and then compared that to a computer module of the same simulation. Then several more real-world phenomena were provided, allowing teachers to see how computer models simulate these phenomena. At the end, teachers learned about connections between science education standards and CS education standards. Scaffolded programming exercises were integrated throughout the whole module.
3. The second module was similar in structure but focused on connecting computer modeling related to math learning.
4. The third module tried to reinforce CS concepts such as procedures, loops, and variables, as well as introduce new concepts like categories of agents. Again, teachers learned how to model various real-world events, ranging from dice rolling to epidemics. For instance, in lesson one, teachers worked on a simple predator-prey model to learn ecosystem dynamics. by using and modifying a computer model of a simple virtual ecosystem.
5. The fourth module asked teachers to start building their own models to illustrate the usefulness of models in different STEM fields. Teachers not only practice CT concepts like operators and sequences, but also exercise CT skills such as testing, debugging, and abstracting using procedures. The first four modules were built on the Project GUTS' PD program funded by National Science Foundation.
6. The fifth module took a different approach, adopting game-based learning instead of focusing on simulation. In this module, teachers consolidated their CT skills through the design and development of their own educational games.

Scaffolding was carefully designed into all the models that moved teachers progressively toward deeper understanding and stronger skill acquisition. For example, the first lesson had no programming requirements but instead asked teachers to examine the existing codes to introduce basic programming concepts and terminologies (e.g., conditionals, Boolean expressions).

When selecting teachers to participate, we intentionally chose two or three teachers from the same school whenever possible, hoping to promote collaborative learning. This project was partially supported by a state grant and the participating teachers were paid for their PD participation.

Participants

Initially, a total of 28 K-12 teachers participated and completed the first round of surveys. However, three teachers discontinued the program after the first session. The remaining 25 teachers constituted the sample of this study.

Amongst the teachers, ten were teaching in elementary, six in middle, eight in high schools, and one was a coach working with K-12 teachers. Five participants taught only math while eight taught sciences exclusively. The remaining 12 teachers taught both math and science. All but two teachers had no prior training in programming.

Data

This study was part of a large research project aiming to help teachers gain content and pedagogical knowledge of CT. The initial data collection included surveys, quizzes, pre-tests and post-tests of CT knowledge and pedagogy, teacher written assignments, researchers' reflective journals, and teachers' feedback through interviews. Teachers' written assignments included modified lesson/unit plans, reflections, and online interactions posted throughout the course and were analyzed using thematic analysis by the researchers. This paper focuses on the impact as well as the efficacy of the described PD model in improving teachers' content knowledge of CT primarily through analysis of the results of the pre- and post- tests related to CT, written assignments, and researchers' reflective journals. The assessment tool developed by the National Science Foundation funded Project GUTS program was used for the pre- and post-tests. The tests consisted of multiple-choice questions assessing teacher's knowledge related to CT. The questions ranged from definitions of CT and modeling, to basic concepts of programming and commands, to connections of modeling to sciences. The appendix provides sample questions.

Analysis

To answer the first research question, descriptive analyses of the 28 participating teachers' pre- and post-tests of content knowledge were conducted first. Due to various reasons, only 17 teachers completed both pre- and post-tests. Their test results were analyzed using a paired t-test.

To answer research questions 2 and 3, thematic analysis of the qualitative data, including the teacher written outcomes and researcher reflective journals, was conducted in an ongoing, iterative fashion. Team members discussed data and informally shared thoughts during project meetings. The formal analysis started after raw data was prepared. First, all data were read through by two researchers, with initial themes identified. A matrix was created summarizing the main topics. Then the preliminary themes were revised by clustering similar themes, and organized as recurring, unique, and leftover themes. Next, the entire data set was recoded, adding new emerging themes, and identified more descriptive wording for the categories. A second matrix was created with all themes included and possible interconnections amongst the themes. Related themes were further grouped together which resulted in the final categories (Creswell, 2014).

Results

Impact on Teachers' CT Understanding

To answer the first research question, pre- and post-tests of teachers' knowledge of computational thinking skills were compared. Twenty-eight participants completed the pre-test and 17 (61% of pretest sample) completed the post-test. Descriptive statistics for each assessment are shown in Table 2. Mean performance increased by over one standard deviation from pre-test to post-test, and the increase in the median score was of similar magnitude. The pre-test mean for the 17 participants who completed both assessments was 57.2 (SD = 14.7), which is similar to the mean from all 28 original participants (M = 58.6, SD = 14.1). The post-test mean was 76.5 (SD = 13.3). A paired t-test was conducted to compare the pre-test and post-test means for the 17 participants with data at both time points. The mean difference was 19.3 (SD = 16.2). The confidence interval for this difference score did not include zero (95% CI: 11.0 – 27.6), and the difference score was significantly different from zero, $t(16) = 4.91$; $p < 0.001$. A 19.3 raw unit difference is 1.31 times the SD of the pretest (Glass' $\Delta = 1.31$), suggesting a large effect size for the increase. That is, the professional development enhanced teachers' understanding of computational thinking as reflected in their achievement test gains.

Table 2

Pre-test and Post-test Descriptive Statistics

		Time	
		Pretest	Posttest
Total correct	Mean	58.6	76.5
	Standard Deviation	14.1	13.3
	Count	28	17
	95% Lower CL for Mean	53.1	69.7
	95% Upper CL for Mean	64.0	83.3
	Median	58.0	75.0
	Mode	58	67*
	Maximum	83	100
	Minimum	33	50
	Range	50	50

Note. CL – Confidence Limits. * = Multiple modes exist. The smallest value is shown.

Integrate CT into lesson planning

To answer the second research question, teacher-developed lesson plans during the summer institute were analyzed. The 17 participating teachers created a total of 14 outcomes ranging from one lesson to a whole unit plan involving eight cycles of lessons. Although the requirement was that everyone needed to produce one lesson plan, many teachers were eager to plan more lessons integrating CT because they saw immense power of such an approach. As a result, a total of eight products went beyond one lesson while six others were plans for a single lesson. Amongst the 14 outcomes, five were at elementary levels and nine were at middle or high school levels. Because collaboration was encouraged, six teachers chose to work with a partner which had resulted in a total of three lessons/units while 11 worked individually. The qualitative analysis of the data revealed three major themes: (a) learner-centred approaches, (b) differentiated learning, and (c) interest in unplugged activities.

Learner-centred approaches. The first common theme identified amongst the lesson plans was that all the teachers adopted learner-centred approaches. Many teachers, at the early stage of the project, were confused about the basic concepts of CT, not to mention how to integrate CT into their teaching practices. After the PD, all the teachers were able to plan at least one lesson applying what was learned during the PD. Although a range of outcomes were produced, a majority of the teachers mentioned how easy it was for them to consider infusing CT into their teaching and many were actually eager to go beyond the requirement and produced outcomes that exceeded our expectations. Chia, a high school science teacher, was very excited to revise a whole unit of a science course she would teach in the coming semester. "I can totally see how CT can be integrated into the course." The unit was called "Earth's Systems: Geological Journeys," and consisted of a total of twenty 60-minute lessons included in eight cycles. Each cycle would start with a real-life scenario where students would act as scholars searching for answers to various essential questions, such as: How can we prevent or reduce the effects of Earth's changes over time? How can natural disasters help to shape Earth's surface? How can patterns in Earth's features help us identify the location of volcanoes? CT concepts and principles were woven into these scenarios where students try to solve these real-life problems.

For example, in cycle three, an existing computer simulation of a real-world phenomenon was introduced for these student scholars to explore. After students had a good understanding of the simulation, they were asked to create a new widget in order to collect data. In cycle five, scholars would play a hurricane simulation to allow them to understand what would happen after a tsunami hits an island and disease begins to affect the population. Next, students were to create their own simulations to model different phenomena. All the projects aimed to allow scholars to comprehend the devastation to our environment, in essence, the course content, while at the time seamlessly integrating CT.

A multidisciplinary approach was also adopted by many teachers. For example, Adam, a high school science teacher, developed two 80-minute lessons focusing on epidemics. The objective of his lessons was "students will be able to model the spread of an epidemic in order to develop a response plan and stop the spread of disease." (Adam's lesson plan). The lessons involved epidemiology, statistics, military biosafety, and governmental travel policy. The unit started with providing context information to students, such as scenarios using general newspaper or magazine clippings showing the beginning stages of disease outbreak. Tying to students' real lives, a video clip related to the recent Ebola outbreak was also shown. Thought-provoking questions (e.g,. What do you notice about people in this video? Why are they so frightened?) were asked, aiming to connect learning with this real-world problem. Then students were asked to generate their own models to simulate the situation in order to discover ways to solve the problem. In the process, various CT concepts were introduced. For instance, to emphasize the concept of abstraction, students were asked to answer questions like: “How is your model accurate? What are somethings that have been simplified/ignored in the model?”

Elementary teachers, just like middle and high school teachers, also took this interdisciplinary path. For example, Beth and Mary designed a whole unit titled: Pirate Treasure for pre-K students. The main objective was to help students learn about patterns, in which big ideas involved counting, quantity, cardinality, patterning, classifying, representing, and interpreting data. The project involved not only math and science, but also social studies and language arts, while at the same time integrating CT concepts like decomposition and algorithm.

Differentiation. Differentiated learning was another common theme identified. Regardless of the content, levels, or the number of the lessons developed, almost all teachers explicitly described how different strategies would be used to meet diverse learner needs.

Warren’s plan provided a good example of such differentiation. His plan integrated CT into middle school math lessons. Focusing on exponential and logistic growth, he infused various CT skills including data analysis, representation, and problem decomposition. The basic activity adapted an existing project where students explore a model of population growth via hands-on activities. Students had to first collect and observe the data without giving any context or reasoning. Then they needed to identify any real scenarios that might be modeled by similar data. Examples of desired answers were epidemics and spread of rumors. Students were then challenged to “develop methods of collecting, storing and analyzing the data using appropriate technological tools and coding” (Warren’ lesson plan). While scaffolding was provided throughout for all students, the level of both mathematics concepts involved and CT skills practiced was different. For example, upper level students were introduced to the logistic curve model and used technology to generate regression equations from the actual data.

Unplugged activities. An intriguing theme that was identified was the use of unplugged activities (i.e., do not involve technology tools) to teach CT. While CT is inherently associated with technology and computers, the teachers seemed to be interested in the idea of using unplugged activities to teach CT concepts. This was the case for all age levels.

The work of Amy and Diane exemplified how unplugged activities could be used to teach CT. They developed a total of nine lessons helping kindergarten students learn the concepts of algorithms. Focusing on the “Plant a Seed” topic, their lessons used daily life activities to teach the content involving CT. For example, after defining the term algorithm,

> Have a class discussion about the steps that students take to get ready for school in the morning. As students are saying things they do to get ready, write them up on the board. Once finished writing out the ideas, put numbers next to their responses to indicate order. Be sure to point out places where order matters and where order does not matter. Explain to students how we just created an algorithm for how they get ready for school in the morning.

In the next activity of the unit, students needed to cut nine squares from a “Plant a Seed” worksheet and determine which six of the nine squares were useful for planting a seed. They would then glue the six pieces in correct order to a sentence strip and swap the sentence strips with other groups. Finally, each group would actually plant a seed following the sequence of steps the other group described in the sentence strip. Apparently, CT skills such as decomposition and abstraction were organically infused into these lessons without using any technology tools.

Several other lesson plans, such as Warren's plan discussed above, either used unplugged activities exclusively or combined both plugged and unplugged practices to teach the content while injecting CT exercises.

CT Integration in Classrooms

The third question sought to answer: How did teachers integrate CT into their STEM teaching as reflected in their daily practice? Data collected at the colloquium and informal interviews of teachers were analyzed to answer this question.

The first theme was that teachers' instructional practice not only built on their own background knowledge, but also was a thoughtful consideration of student interests and content needing to be taught. Teachers carefully embedded relevant CT concepts into the subject matter knowledge. Jane was teaching high school science at an online school. She initially designed a 90-minute lesson asking students to "[re-evaluate] based on data collected from a lab…reconstruct a model based on collected data." When she actually implemented the lessons, she expanded the lessons to a whole unit. In her unit plan, the task set for students was as follows:

> You have learned that clean, fresh water is a limited resource that both humans and other organisms depend on for survival. With a growing population, it is important to consider ways to reduce water consumption. Although you could estimate water savings by changing some behaviours at home, developing and using a mathematical model will allow you to calculate *actual* water use reductions. In this task, you will examine your household water usage in order to develop an equation to model, analyze, and design a plan to reduce your home water consumption by 25%. Throughout this task, you will use Chart 1 to organize information.

Students were asked to develop a mathematical equation and model to calculate their daily water usage, but also calculate a 25% water reduction plans. Another high school science teacher, Esther, talked about how her practices had changed, "student choice now always includes coding simulation option." She discussed the impact of this PD process:

> I feel the [PD] vastly improved my understanding of CT and ways to incorporate it into lessons, because with clear understanding of the concepts, it is much easier to incorporate it. This is something that a lot of science teachers could benefit from…It helped me to learn and implement more CT in my Next Generation Science Standards (NGSS) (Next Generation Science Standards) lessons. Most teachers do not yet incorporate these skills. In fact, I have suggested this training be included in county PD." (Esther, colloquium)

During the Summer Institute, Peter developed plans for two 80-minute science classes aiming to enhance his high school students' understanding of epidemics through modifying existing computer simulated models. The practices emphasized two important CT concepts: decomposition and abstraction. For example, the question "How is this model accurate and what are some things that have been simplified/ignored in the model?" [Peter, lesson plan] evidently tried to guide students' thinking about these topics.

Five months later at the colloquium, Peter's presentation of what he had accomplished floored the audience. Collaborating with another participating teacher who worked at the same school, he and his collaborator founded a STEAM Career Exploration club in their school attracting about 50 student members. In that club, students worked on various projects including, but not limited to: Cyanotopys, Ozzo-Blockly, and building and design with Makedo and TinkerCAD. Over 100 students successfully completed an “Hour of Code” project in which they had to not only complete the Hour of Code, but also write about the connection between coding to a science field. Peter carefully considered three critical components: his background (his master’s thesis focused on health-related science education), students’ interest in prosthetics, and the course he was teaching. He revised his plans and shifted the focus from epidemics to prosthetics. The initial problem he had was that “prosthetics are costly and out of research for many children around the world. It costs between $50,000 to $100,000 and is easily damaged. Kids also outgrow them in about a year.” They found an open source Computer Aided Design (CAD) software that provided a solution where people can design prosthetic limbs, printing with 3D printers. Inspired, Peter and colleagues wrote and won multiple grants to purchase a 3D printer and all of the supplies. Excited students engaged in the authentic experiences of designing prosthetic limbs, 3D printed and assembled them, and later shipping them to amputees around the world.

Students’ motivation was another aspect these teachers found interesting when they implemented the lessons. They found CT became a motivational factor for students’ learning of the subjects. For instance, Esther compared CT integration to other motivational efforts and concluded that CT increased student engagement in various activities each time. She looked at both extrinsic and intrinsic motivations. In her lessons, extrinsic motivators included awarding points for completion of tasks and giving prizes for good performance, while intrinsic motivators were allowing students to work on optional coding/simulation projects. She found that giving students options to work on coding projects largely motivated students’ completion of the tasks. In particular, it was excellent in helping students with varied language abilities. During her presentation at the colloquium, she stated: “I have a lot of ESOL students, [CT] greatly enhanced their performance, they are able to incorporate their skills and apply their CT strategies and design process to be successful, which help them be even more motivated.”

Lessons Learned

The fourth research question focused on the lessons learned from the implementation of the PD. Thematic analysis of the data, including teachers’ feedback from the surveys and researchers’ notes, was conducted to answer this research question. Several important lessons were identified through this analysis.

First, the blended learning approach afforded teachers flexibility to learn anytime and anywhere. On the one hand, the online modules offered much-needed flexibility. Practicing teachers are very busy, especially during the school year, and the online learning modules allowed participants in the study to learn at convenient times and places. In addition, the online learning environment enabled teachers to access global resources. For example, two of our CT experts participated from different physical locations, one outside the country. Yet interaction with experts both local and remote was easy and accessible. Also, the online learning experience allowed self-pacing for both slow and quick learners, making learning less stressful and more enjoyable. The embedded assessment tools such as quizzes not only provided immediate feedback so that teachers could better monitor their learning, but also enabled easy tracking of teachers’ learning progress. On the other hand, the face-to-face elements provided teachers with opportunities to establish collaborative relationships. The first face-to-face orientation meeting allowed teachers to bond quickly and also provided an opportunity for the teachers to acclimate to the format of the modules. The integrated optional technical support sessions afforded chances to get just-in-time help for those teachers who were not confident in their skills and needed additional assistance. The face-to-face summer institute facilitated more collaboration amongst teachers. It also empowered teachers to engage in deep learning that involved more hands-on, minds-on activities.

In addition, a few challenges were identified. One significant challenge was how to develop learning materials that could meet the needs of diverse teachers. While it was our intention to include teachers of grades K-12, so that different teacher groups could learn from each other, we found that it was challenging to create opportunities that could simultaneously foster both elementary and secondary teachers' learning experiences. This is reflected in two perspectives: first, secondary teachers tended to have stronger STEM backgrounds than elementary (especially pre-K to second grade) teachers. Second, middle, and high school teachers were more likely to find connecting CT to their curriculum easier than their colleagues teaching in elementary. In particular, it was difficult for few pre-K to second grade teachers to consider how CT could be integrated into their classrooms.

Balancing the face-to-face and online meeting times presented another challenge. A vast majority of the participating teachers enjoyed the online, asynchronous learning approach, especially during the regular school year because it offered them the needed flexibility to work on the materials at the time and place they preferred. Some teachers even stated that “I would not be able to participate if [the modules] were not online.” However, a few teachers were frustrated because they needed more intensive technical support. They suggested including a few face-to-face sessions, especially at the beginning of the project, to support the steep learning curve of programming. Our participants’ thoughts are consistent with findings from the case study (Israel et al., 2015) in which teachers observed that a lack of computing expertise was a significant barrier and that ongoing support was a significant aid to teachers’ integration of CT into the classroom.

Discussion

The importance of CT has been increasingly realized and consequently it is critical to equip teachers with the content and pedagogical knowledge related to CT integration. As suggested by various studies (Voogt et al., 2015), a much-needed area of research is how to integrate CT in education, and in particular, how teachers can incorporate CT in disciplines other than computer science. The results from this study shed light on this field.

First, the quantitative result shows positive outcomes of this PD in helping teachers learn a set of CT skills. This finding is similar to the work of Bower and colleagues (Bower et al., 2017), who demonstrated that teachers improved their confidence and attitudes toward teaching CT, as well as awareness of technology and resources, following a brief PD workshop. In our study, teachers have gained CT knowledge after a more extensive PD experience.

Secondly, this study of teachers with no prior experience in computer science highlights how CT concepts and ideas can be incorporated into their daily practices. CS is a challenging subject, which is partially evidenced by the high dropout rate of students. Similarly, teachers, especially those without much background in CS, can have the same fear about this subject and give up easily. Yet, the results of this study show that when concrete examples of CT from day-to-day life, along with terminologies, are woven into daily life scenarios, teachers are able to acquire the content knowledge of CT, and at the same time are able to consider blending CT with different subjects. This confirms previous research (Yadav et al., 2014).

The finding that teachers are interested in and have integrated unplugged activities in their lessons suggests another great way to expose students to CT. Different from other subjects like math or language arts, CT and CS are usually, if not always, learned in a computer environment. This could create a barrier for some students (likely a large portion of our students) who dislike computers due to various reasons ranging from fear, personal preferences, or lack of opportunity to learn. Grounded in enactivism, the unplugged ways to teach CT can not only provide learners an intuitive approach to learn CT but also will open a gateway to the fascinating world of CS. Stressing embodied cognition with physical manipulatives, unplugged activities have been intentionally built into the PD. This has allowed teachers to learn CT in rich and authentic ways by placing them at the centre of the learning process while incorporating bodily movements. It is exciting to observe that they are able to transform such knowledge for their own practice.

Third, the analyses of teacher-developed teaching plans show that all the teachers adopt the learner centred approach, especially multidisciplinary approaches, regardless of the content or grade level of the lessons. Such unanimous adoption is not expected because traditional ways of teaching are still common practice in subjects like math and science, especially in middle and high school classrooms (Ferguson, 2010; Stipek, Givvin, Salmon, & MacGyvers, 2001). A plausible explanation is that "teachers teach the way they are taught," since this PD adopted a learner-centred approach, in which teachers learned through interdisciplinary projects embedded in real life situations. Another important factor is that CS, through which CT is derived, is a field that lies on the intersection of so many other subjects such as mathematics, circuit design, software development, data sciences, and engineering. For example, modeling, a key CT topic, is inherently connected to a number of fields. It is therefore logical for the teachers to adopt multidisciplinary approach in their plans. We applaud that the teachers are able to weave CT skills into the subjects other than CS, aiming to help students "understand how to use computation to solve their problems ... to discover new questions that can fruitfully be explored" (Voogt et al., 2015 p. 725).

The topic of extensive PD is needed and is worthy of further discussion because it also entails cost and time. As discussed in the professional development section earlier, the comprehensive review of existing literature and an evaluation of a national survey (Desimone & Garet, 2015) demonstrate that extensive, supported PD is essential. Consistent with the existing literature and PD models, our PD has been long-term in that we heavily invested in these teachers' successes. In addition to the online modules and face-to-face PD sessions, this investment also includes continued technical support as well as pedagogical support from three content experts who have 20+ years of experience in the respective content areas. Additionally, teachers were compensated for their participation. While much support and time were provided to help teachers gain both the content and pedagogical knowledge to integrate CT, a number of teachers still dropped out. One reason for discontinuing their participation, according to a couple of the teachers, is that they became too busy. Another possible cause may be the lack of prior knowledge in programming. CT is very challenging for teachers to learn for different reasons. First, learning CT through coding is similar to learning a new language that is best learned when one is immersed in this language. Teachers typically do not have the chance to be immersed in this "language" (i.e., coding) environment. Secondly, CT learning requires rich hands-on practices where one learns through creation of products rather than just talking/reading about it. This demands patience, time, and effort. Thirdly, CT is abstract and considered to be at the highest level of thinking skills. It calls for one to communicate and solve problems with a machine, which is very different from the ways we humans typically approach problems and problem solving. Understandably, abstract thinking is difficult. Considering these barriers, it is not surprising that some teachers unfortunately gave up.

Conclusion

The idea of helping all students develop computational thinking skills has gained momentum in recent years. While much effort has been made, including large scale initiatives in research and development, research on CT integration in subjects beyond computer science is still lacking (Voogt et al., 2015) and even less attention has been paid to the training of teachers for such integration. This paper explores in-service teacher training and has identified positive practices that can provide useful information. It also deepens our understanding of teacher beliefs and their ways of CT integration into their daily practices. The lessons learned from this work allow us to make the following recommendations when considering future PD related to CT integration.

1. Any CT focused PD should be offered over an extended period of time instead of only once or twice. While it is useful to offer a few intensive PD experiences (e.g., like the summer institute we had), teachers need time to digest the knowledge learned and opportunities to apply and explore. This is particularly true for CT learning since it needs ample hands-on opportunities to practice before one can master it.
2. Blended learning for CT focused PD proves to work well for the teachers who are busy, especially during the school year. However, how to balance the online and face-to-face meetings is key. Our results show that teachers welcome 65% or more online PD sessions but need a few face-to-face optional tutorial sessions throughout, especially for those with weak technical backgrounds. Considering the unique characteristics of CT and teachers' possible apprehension associated with programming, it is desirable to embed unplugged activities throughout PD. We suggest starting the PD with a few face-to-face sessions in which unplugged activities are integrated to alleviate fear of coding. These sessions should also introduce basic programming with ample technical support to allow teachers, especially those who are beginners, to smoothly launch their journey of coding.
3. Collaboration is another factor that contributes to success of PD, particularly for CT learning. In this study, whenever possible, we intentionally selected teachers from same schools and encouraged collaborative practices. We have found that this benefits the teachers in two ways: first, they are likely to be persevere and be able to complete all tasks. As discussed earlier, CT is challenging and having peer collaborators allows teachers to build confidence and learn from each other. Working together can also push teachers out of their comfort zone and encourage them to take risks. Last, the lesson plans developed collaboratively tend to be more comprehensive because teachers have the opportunity to look at CT from different perspectives.

Acknowledgements:

This research was partially supported by an Improving Teacher Quality grant from the Maryland Higher Education Commission (MHEC), USA. However, the views and findings expressed here are of the authors and do not necessarily reflect the views or positions of MHEC. The authors want to thank Dr. Eric Klopfer and Irene Lee for their support and permission to adapt the Project GUTS curriculum. They also wish to thank all the participating teachers as well as to Arkhadi Pustaka and Wendy Huang for their assistance.

References

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.

Blanchard, M. R., LePrevost, C. E., Tolin, A. D., & Gutierrez, K. S. (2016). Investigating technology-enhanced teacher professional development in rural, high-poverty middle schools. *Educational Researcher, 45*(3), 207-220.

Bower, M., & Falkner, K. (2015). *Computational thinking, the notional machine, pre-service teachers, and research opportunities.* Paper presented at the Proceedings of the 17th Australasian Computing Education Conference (ACE 2015).

Bower, M., Wood, L. N., Lai, J. W., Howe, C., Lister, R., Mason, R., Highfield, K., & Veal, J. (2017). Improving the computational thinking pedagogical capabilities of school teachers. *Australian Journal of Teacher Education, 42*(3), 4.

Creswell, J. W. (2014). *Research design: Qualitative, quantitative, and mixed methods approaches* (4th ed.). Thousand Oaks, CA: Sage.

Czerkawski, B. C., & Lyman, E. W. (2015). Exploring issues about computational thinking in higher education. *TechTrends, 59*(2), 57-65.

Davis, B., Sumara, D., & Luce-Kapler, R. (2008). *Engaging minds: Changing teaching in complex times* (2nd ed.). Mahwah, NJ: Erlbaum.

Desimone, L. M., & Garet, M. S. (2015). Best practices in teachers' professional development in the United States. *Psychology, Society and Education, 7*(3), 252-263.

Fenwick, T. (2000). Expanding conceptions of experiential learning: A review of five contemporary perspectives. *Adult Education Quarterly, 50*(4), 243-272.

Ferguson, K. (2010). *Inquiry Based Mathematics Instruction Versus Traditional Mathematics Instruction: The Effect on Student Unverstanding and Comprehension in an Eigth Grade Pre-Algebra Classroom.* (Master), Cedarville University, Retrieved from https://digitalcommons.cedarville.edu/education_theses/26/

Gadanidis, G., Cendros, R., Floyd, L., & Namukasa, I. (2017). Computational thinking in mathematics teacher education. *Contemporary Issues in Technology and Teacher Education, 17*(4), 458-477.

Grover, S., & Pea, R. (2013). Computational thinking in K–12: A review of the state of the field. *Educational Researcher, 42*(1), 38-43. doi:10.3102/0013189X12463051

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 & Education, 82*, 263-279.

Kell, M., Rupley, W., Nichols, J., Nichols, W., Paige, D., & Rasinski, T. (2016). Teachers' Perceptions of Engagement and Effectiveness of School Community Partnerships: NASA's Online STEM Professional Development. *Journal of Studies in Education, 6*(2), 1-23. doi:10.5296/jse.v6i2.9185

Li, Q. (2014). *Learning Through Digital Game Design and Building in A Participatory Culture: An Enactivist Approach*. New York, NY: Peter Lang.

Merleau-Ponty, M. (1964). *The primacy of perception and other essays*. Evanston, IL: Northwestern University Press.

Mouza, C. (2009). Does research-based professional development make a difference? A longitudinal investigation of teacher learning in technology integration. *Teachers College Record, 111*(5), 1195-1241.

Nardelli, E. (2019). Do we really need computational thinking? *Communications of the ACM, 62*(2), 32-35.

Oluk, A., & Korkmaz, O. (2016). Comparing Students' Scratch Skills with Their Computational Thinking Skills in Terms of Different Variables. *International Journal of Modern Education and Computer Science, 8*(11), 1-7. doi:10.5815/ijmecs.2016.11.01

Reid, D. (1995). *The need to prove.* (Ph. D doctoral dissertation), University of Alberta, Edmonton.

Stipek, D. J., Givvin, K. B., Salmon, J. M., & MacGyvers, V. L. (2001). Teachers' beliefs and practices related to mathematics instruction. *Teaching and Teacher Education, 17*(2), 213-226.

Varela, F., Thompson, E., & Rosch, E. (1991). *The embodied mind: Cognitive science and human experience*. Cambridge, MA: Massachusetts Institute of Technology Press.

Voogt, J., Fisser, P., Good, J., Mishra, P., & Yadav, A. (2015). Computational thinking in compulsory education: Towards an agenda for research and practice. *Education and Information Technologies, 20*(4), 715-728. doi:10.1007/s10639-015-9412-6

Williams, H. (2017). *No Fear Coding: Computational Thinking across the k-5 Curriculum*. Arlington, VA: International Society for Technology in Education.

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 (TOCE), 14*(1), 5.

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

Appendix: Pre-Test and Post-Test Sample Questions

1. Computer modeling is used to study problems that are:
    a. Too dangerous to study in the real world
    b. Too time consuming to study in the real world
    c. Too expensive to study in the real world
    d. All of the above
2. In a computer program, a loop is used…
    a. To jump around in code
    b. To perform simultaneous tasks
    c. To store information
    d. To repeat a set of instructions
3. For a model to be useful it must be:
    a. As realistic as possible
    b. Predictive
    c. Validated by data
    d. None of the above

Authors

Sarah Haines: Dr. Sarah Haines is a full professor of biology and science education at Towson University, Towson, Maryland. She earned an M.S. and PhD in Zoology from the University of Georgia. She taught seventh grade science before coming to Towson University. Her research interests lie in the areas of environmental education and outdoor education, and how they affect student achievement. Email: shaines@towson.edu

Qing Li: Dr. Qing Li is a full professor of educational technology at Towson University, US. Her current research interests include elearning, game based learning, computing education, cyberbullying and educational technology. She has published widely in these fields, including over 60 peer reviewed journal articles, two books, and numerous conference papers. Dr. Li was a visiting scholar at the Massachusetts Institute of Technology (MIT) in 2009 and a visiting professor at the University of Idaho in 2006. Email: li@towson.edu

Scot McNary: Dr. Scot W. McNary is Associate Professor in the Department of Educational Technology and Literacy at Towson University and teaches courses in research design, statistics, and research proposal writing. His research interests include quantitative methods and program evaluation. Email: SMcnary@towson.edu

Laila Richman: Dr. Laila Richman is an Associate Professor and the Associate Dean for the College of Education at Towson University. She received her Ph.D. in Special Education at the University of Kansas and, prior to completing her doctoral program, taught eighth grade special education. Her research interests and scholarship focus on Universal Design for Learning, instructional technology, and teacher preparation. Email: lrichman@towson.edu