Examining Pre-service Teacher Knowledge Trajectories of Computational Thinking Through a Redesigned Educational Technology Course

Hui Yang, Chrystalla Mouza, Yi-Cheng Pan,
Email: huiy@udel.edu, cmouza@udel.edu, ethanpan@udel.edu
University of Delaware

Abstract: Computation Thinking (CT) is a fundamental skill of analytical thinking. To successfully infuse CT into K-8 settings, we must equip pre-service teachers with CT knowledge and skills that can be applied in their curricular context. This work presents trajectories of pre-service teacher knowledge development of CT concepts and computing tools within the context of disciplinary content and pedagogical knowledge through their participation in an educational technology course. Data were collected from a random sample of 42 pre-service teachers. Data sources included participants' lesson plans (N=126) collected through three different time points. These materials were analyzed quantitatively and qualitatively. Preliminary results indicated that pre-service teachers gradually improved their understanding throughout their participation in the course. Findings also revealed difficulties in conceptualizing and integrating CT in conjunction with content and pedagogy, particularly during authentic vis-à-vis hypothetical lesson designs.

Introduction

In recent years, researchers and policy makers have established a compelling rationale for introducing computing in K-8 contexts as a means of advancing student development of Computational Thinking (CT). Broadly speaking, CT is a problem-solving methodology that can be implemented with a computer and can be automated, transferred, and applied across subjects (Barr & Stephenson, 2011). Wing (2006) suggests that CT is "a universally applicable attitude and skillset for everyone, not just computer scientists …" (p. 33). Accordingly, the International Society for Technology in Education (ISTE) in collaboration with the Computer Science Teacher Association (CSTA) identified CT concepts for K-12, which include (a) problem decomposition as "breaking down complex problems into more manageable parts"; (b) algorithmic thinking as "using a precise sequence of steps or instructions to solve problems"; and (c) simulation as "representing a process" (CSTA & ISTE, 2011). Indeed, the newly released National Educational Technology Standards for Students emphasized the need for advancing our younger generations to develop CT skills required to navigate in the digital world (ISTE, 2016). Promoting the development of CT knowledge and skills in K-8 settings, however, is challenging because teacher preparedness is a major barrier (Code.org, 2017). Thus, a critical step for successfully infusing CT into K-8 classrooms is to help pre-service teachers build an understanding of CT and its connection to their curricular context (Yadav, Hong, & Stephenson, 2016; Yadav, Stephenson, & Hong, 2017).

Since its inception, the framework of technological pedagogical content knowledge (TPACK) has provided a unifying lens for researchers working to explicate teacher knowledge for effective use of technology tools and practices across the curriculum (Mishra & Koehler, 2006). Thus, TPACK provides a useful framework for studying teacher knowledge in relation to CT, because computational tools are central to CT (Angeli, Voogt, Fluck, Webb, Cox, Malyn-Smith, & Zagami, 2016). Research that examines TPACK in relation to CT, however, is only now beginning to emerge (Mouza, Yang, Pan, Ozden, & Pollock, 2017). One way to advance pre-service teacher preparation for integrating CT into K-8 education is through stand-alone educational technology courses required in most teacher education programs around the U.S. (Yadav, Gretter, Good, & McLean, 2017). In this work we present the design of an educational technology course which introduces computing tools, vocabulary, and practices specific to incorporating CT within the context of content and pedagogical knowledge in K-8 settings. Relatedly, we explore the following research question: *What learning trajectories are exhibited by pre-service teachers as they learn and apply CT-related concepts and computing tools within the context of disciplinary content and pedagogical knowledge?*

Theoretical framework

This work is situated in the theoretical framework of TPACK (Mishra & Koehler, 2006). Building upon Shulman's (1987) scholarship of teacher knowledge, TPACK centers on the nuanced interactions among three bodies of knowledge (Figure 1): content knowledge (CK), technology knowledge (TK), and pedagogical knowledge (PK). CK refers to knowledge of subject matter. TK refers to knowledge of various technologies and appropriate vocabulary (e.g., terminology). PK refers to knowledge of methods and processes for teaching. These domains combine to form three additional constructs. Pedagogical content knowledge (PCK) refers to knowledge of representing content to make it comprehensible to others. Technological content knowledge (TCK) refers to knowledge of how technology can create new content representations. Technological pedagogical knowledge (TPK) refers to knowledge of how various technologies can be used in teaching. When technology, content and pedagogy blend together, the result is TPACK– a synthesized form of knowledge that supports effective use of technology within specific subject domains.

In prior work (Mouza et al., 2017), we explicated the construct of TPACK in relation to CT, focusing on what all teachers need to know and be able to do in order to use CT as a means for exploring disciplinary content (e.g., math, science, literacy, etc.). Towards this end we advanced the term of TPACK-CT, which focuses on pre-service teachers' ability to understand how CT-related concepts, computing tools, and practices (TK) can be combined with disciplinary content (CK) and pedagogical strategies (PK) to promote meaningful student outcomes in specific contexts (see Figure 1). The construct of TPACK is used in two ways in this work: as a framework guiding the design of the educational technology course, and as an analytic lens for examining pre-service teacher outcomes as illustrated in course products. We focus exclusively on the construct of TPACK-CT, because our goal is to move beyond the individual knowledge components of technology, content and pedagogy to illustrate a synthesized form of knowledge that supports effective use of CT-related concepts and tools within specific subject domains.

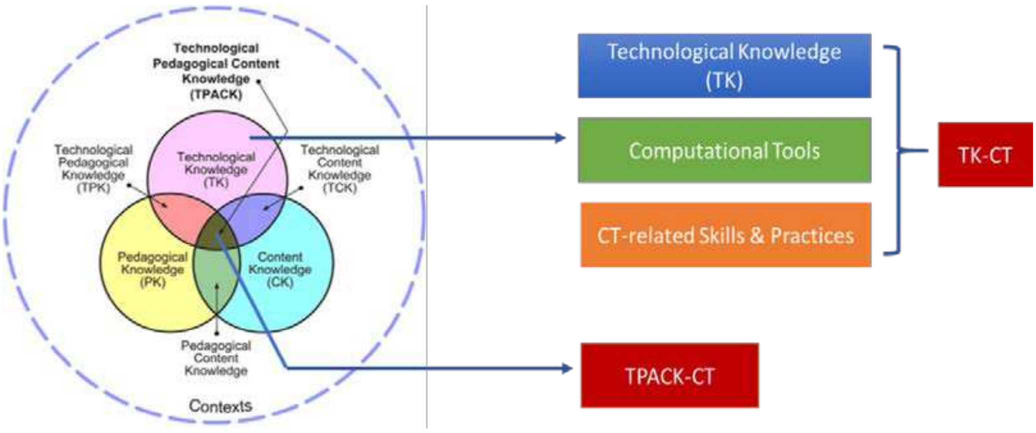

Figure 1. TPACK-CT framework (based on Mishra & Koehler, 2006).

Methods

Context

This study was conducted in the context of a four-year undergraduate teacher education program in the United States. Graduates of the program are eligible for both elementary (K-5) and middle school (6-8) teacher certification. The program curriculum is divided into three areas: (a) the *general studies* courses which help develop subject matter knowledge; (b) the *professional studies* courses (e.g., methods) which prepare pre-service teachers for their future classroom; and (c) the *concentration* courses which help develop expertise in a middle school content area. Additionally, the program curriculum is designed to provide pre-service teachers with a range of field experiences in a variety of classroom settings. These experiences culminate with student teaching.

Course description

Integrating Technology in Education is a 15-week course required for all pre-service teachers during their junior or senior year. Typically, the course introduces participants to technologies available for use in classroom content areas, pedagogical considerations with these technologies, and teaching and learning practices that combine the use of technologies with content and pedagogy. The specific tools used in the course change frequently to keep pace with rapid advances in technology, but typically include tools that support communication, content representation, collaboration and production. Concurrent with the course, pre-service teachers complete methods courses and accompanied field experience for 3 full weeks within a classroom setting. The field experience allows the opportunity to engage in the design and application of authentic classroom materials that embed technologies in the context of content area instruction. For the purpose of this work, we redesigned this course to support the development of pre-service teachers' use of CT-related concepts and computing tools (TK-CT), as well as to promote their CT practices within specific disciplinary and pedagogy (TPACK-CT). Table 1 provides an overview of the course design. The course is offered every semester but for the purpose of this work we focus on all sections offered during three consecutive semesters (Spring 2016, Fall 2016 and Spring 2017). All sections were taught by the same instructor (second author) and utilized the same syllabus and course activities.

Table 1: Description of the CT-infused educational technology course

Technology	Activity	CT Supported Skills
Interactive Whiteboards	Identify two interactive whiteboard resources that support key CT skills: *Modeling* (e.g., a resource that can be used to represent a phenomenon such as prey and predator relationship); *Sequencing* (a resource that could be used to sequence events); *Data* (e.g., a resource that could be used to represent data such as a graph); and *Sorting* (e.g., a resource that could be used to organize information)	Modeling Abstraction Algorithmic Thinking Data & Sorting
Programming Hour of Code Scratch	*CS unplugged* (activity done without computers to introduce algorithms) *Hour of Code*: Completion of a Grades 2-8 activity *Introduction to Scratch Programming*: Scratch is an object oriented programming language Review lessons that support the use of Scratch: ScratchED Design of a learning activity in a content area that involves Scratch Programming/Reflection	Algorithmic Thinking Problem Decomposition Modeling; Abstraction
Concept Mapping Tools	Design of a learning activity that uses concept mapping in a content area to support student development of CT skills, such as decompose a mathematical problem, model abstraction (e.g., life cycle of a butterfly), sequence events in a story or plan essay execution.	Problem Decomposition Algorithmic Thinking Modeling Abstraction
Data	Introduction to Internet Research including use of keywords, boolean logic and operators and evaluation of online content	Problem Decomposition Modeling Abstraction
Collaboration tools	Select and read an article on multiple approaches to developing CT: board games, robotics, programming. Use of a collaboration tool to present the reading to classmates.	Problem Decomposition Algorithmic Thinking Modeling Abstraction

Participants

A total of 135 pre-service teachers enrolled in six sections of the course between Spring 2016-Spring 2017. For the purpose of this study, six to eight pre-service teachers were randomly selected from each section for a total of 42 participants (N=42), which represents approximately 30% of all participants.

Data collection

To examine pre-service teachers' growing trajectories as they represented and applied their TK-CT with content and pedagogy in the context of classroom teaching (TPACK-CT), we collected course materials at three different time points. For time point 1 (T1), we collected pre-service teachers' lesson plans incorporating a programming tool – Scratch (see Table 1). In this activity, pre-service teachers developed lesson plans that incorporated Scratch programming within a curricular content area. To accomplish this goal, they first examined user-created projects available on the Scratch community that fit their curricular goals. Subsequently, they examined a variety of lesson plans through the ScratchEd community, an online publicly available forum where educators exchange resources, pose questions and share experiences to broaden the integration of Scratch into core curricular contexts. Finally, they identified a learning goal within a content area of their choice (e.g., science, mathematics, social studies, English) and developed a lesson plan that integrates programming with curriculum content. To scaffold lesson development, pre-service teachers were provided with a series of prompts aligned with the TPACK framework (see Harris, Grandgennet, & Hofer, 2010), including (a) *planning* (e.g., Consider the pedagogical decisions you'll need to make to develop this lesson idea. How will you introduce Scratch to your students? What activity types will students engage with to learn the concept? In what ways will you assess student learning?); and (b) *reflection* (e.g., How will programming help your students achieve the identified learning goal? How will the lesson support the development of students' CT skills? What was your experience with Scratch?).

For time point 2 (T2), we collected pre-service teachers' lesson plans incorporating a concept mapping tool. Initially, all pre-service teachers were introduced to various types of concept mapping tools. Much like learning to use programming tools, pre-service teachers were subsequently asked to identify a learning goal within a content area of their choice (e.g., science, mathematics, social studies, English) and develop a lesson plan that integrates concept mapping in a core curricular area. To scaffold lesson development, a series of planning and reflection prompts was again provided that paralleled those of the programming activity.

For time point 3 (T3), we collected pre-service teachers' final course products completed through a case development project. The case development project progressed incrementally through stages that allowed participants to implement and reflect on their own lessons that supported the development of CT knowledge and skills among school students. Participants were allowed to implement one of the lessons designed throughout their participation in the Technology Integration course (e.g., programming, concept mapping) or design a new lesson that integrated technology with content and pedagogy aimed at supporting students' CT knowledge and skills. The culminating component of the project was a reflective case report of approximately 1,000 words written in response to several prompts. Specifically, each case report was divided into two sections: (a) *case narrative* (e.g., How did you introduce the lesson to students? What happened during the actual implementation of your lesson?), and (b) case reflection (e.g., How did the lesson support the development of students' CT skills? What are two things you will remember about this lesson for future planning?).

It is important to note that the programming and concept mapping activities asked pre-service teachers to develop hypothetical lesson plans, which helped them envision the infusion of CT into their future classroom. In contrast, the final case development activity asked participants to implement and reflect upon an authentic lesson designed and enacted in their field experience. All assignments followed a consistent format; they first engaged pre-service teachers in developing knowledge and skills related to CT concepts and computing (TK-CT) and subsequently asked them to apply such knowledge in the context of disciplinary content and pedagogy (TPACK-CT).

Data analysis

A total of one hundred and twenty-six lesson plans (N=126) were scored using a modified version of the *Technology Integration Assessment Rubric*, a valid and reliable instrument aligned with the TPACK framework, that can be used to evaluate pre-service teachers' lesson plans (Harris, Grandgenett, & Hofer, 2010). The rubric identifies four evaluation criteria which include: (a) curriculum goals and technologies (e.g., computing tools and practices that support the development of CT knowledge and skills); (b) instructional strategies and technology: using computing tools to support teaching and learning that fosters students' CT knowledge and skills; (c) technology selection(s): compatibility with curriculum goals and instructional strategies; and (d) fit: alignment of content, pedagogy, and computing tools to foster CT knowledge and skills. Each of the four criteria can receive a numerical score from 1 to 4. A score of 1 indicates failure in satisfying the criterion, while a score of 4 indicates full success in satisfying the criterion. Each lesson plan was scored by two researchers. The initial inter-rater reliability was calculated at 87%. All discrepancies were discussed until a 100% agreement was reached.

A one-way repeated measures analysis of variance (ANOVA) was conducted to analyze and identify all 42 participants' TPACK-CT growing trajectory based on their overall scores at three different time points (T1, T2 and T3). Subsequently, participants' T1 overall scores were categorized in three groups: High Start, Low Start and Medium Start groups. The High Start group (N=5) included participants whose lesson plans scored 3.0 or above (top quarter). The Low Start group (N=24) included participants with lesson plans that scored below 2.0 (bottom half). The Medium Start group included participants (N=13) whose lesson plans received scores between 2.1 to 2.9. Based on initial scores, data were subsequently analyzed using a two-way repeated measures analysis of variance (ANOVA) to present growth trajectories among the three groups, followed by a one-way ANOVA performed at each time point to determine the mean differences among groups. We subsequently selected three participants, who were representative of each group, and conducted a qualitative analysis of their data entries over time in order to better illustrate how TPACK-CT was represented in these growing trajectories.

Findings

Pre-service teachers' development of TPACK-CT

A one-way repeated measures ANOVA with a Greenhouse-Geisser correction determined that means of participants' overall TPACK-CT scores significantly differed between time points ($F(1.96, 80.52) = 26.97$, $P < 0.0005$). Post hoc tests using the Bonferroni correction revealed that participants' overall TPACK-CT scores significantly increased in the concept mapping lessons (T2) and final case narratives (T3) compared with the Scratch programming design (T1). However, there was a slight reduction, 0.39 point, in participants' TPACK-CT scores from their concept mapping lessons to the final case narrative, which was statistically significant ($p = .03$). Figure 2 provides an overall growing trajectory among all participants.

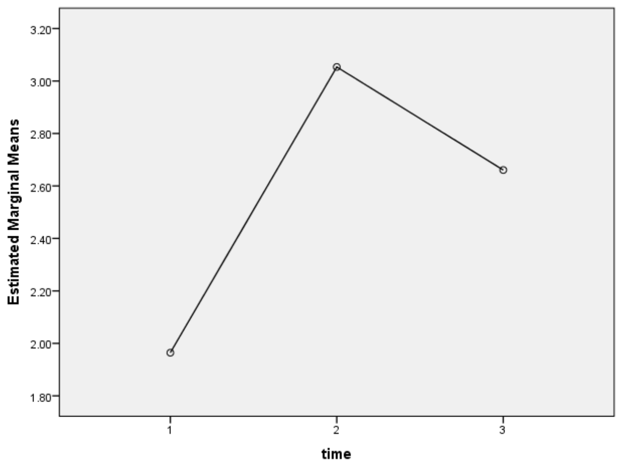

Figure 2. Plot of marginal means across time (N=42).

Results from two-way repeated measures ANOVA revealed that the group-by-time interaction was also significant (Wilks' $\lambda = .42$, $F(2, 4) = 10.18$, $p = .000$). Tukey post hoc tests from one-way ANOVA revealed that a statistically significant difference existed among "High Start", "Medium Start" and "Low Start" at T1 ($F(2, 39) = 168.16$, $p = .000$); however, no significant differences existed among these groups in the concept mapping assignment (T2) and final case narrative (T3) even though they started with significant differences in the programming lesson plan (T1). Table 2 presents the means and standard deviations for the three groups on the dependent variable separated by time period. Figure 3 provides a visual representation of the three growing trajectory patterns.

Table 2: Means and standard deviations by groups and times

Time	Groups	Mean	Std. Deviation
T1	High Start	3.25	0.18
	Medium Start	2.46	0.36
	Low Start	1.43	0.14
T2	High Start	3.20	0.54
	Medium Start	2.86	0.82
	Low Start	3.13	0.69
T3	High Start	2.83	0.73
	Medium Start	2.77	0.66
	Low Start	2.57	0.75

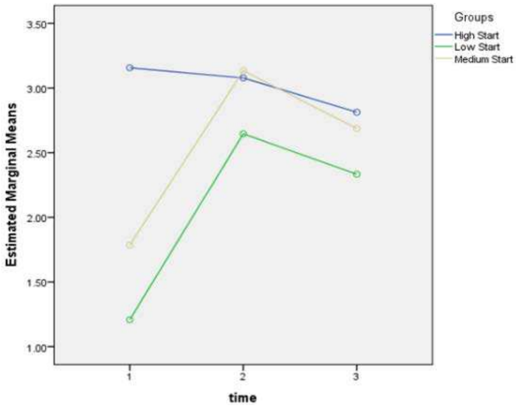

Figure 3. Plot of marginal means by group across time.

Overall, quantitative data indicated a growing understanding of TPACK-CT in pre-service teachers' lesson plans over time. Specifically, participants who started with a high score on the programming lesson plan maintained high TPACK-CT development. Further, participants who had limited to moderate TPACK-CT understanding in the programming lesson plan gradually improved in their development of TPACK-CT as evidenced in the concept mapping and final case narrative. These findings indicate that participation in the Technology Integration course positively influenced pre-service teachers' TPACK-CT. However, as shown in our analysis, participants exhibited some difficulty in articulating TPACK-CT in authentic settings (i.e., case narrative) compared to hypothetical lesson design (i.e., concept mapping).

Pre-service teachers' TPACK-CT representation

To more clearly illustrated pre-service teachers' growing trajectories, we present a more detailed analysis from three participants who represent the growth exhibited within each group: Jim (High Start group), Casie (Low Start group) and Ella (Medium Start group). Figure 4 presents the TPACK-CT growing trajectories of these three participants.

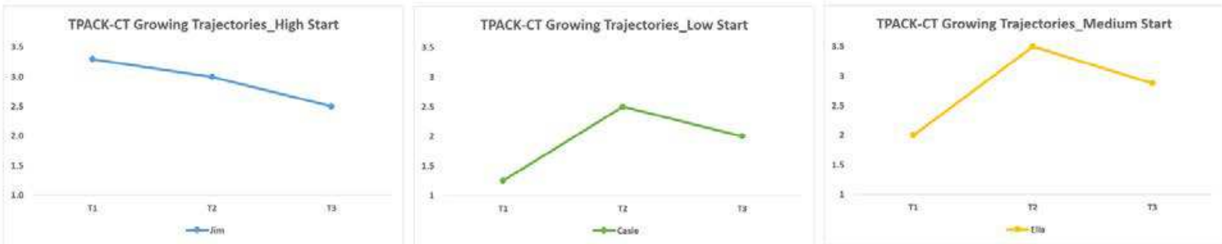

Figure 4. Participants' TPACK-CT Trajectories (Jim – High Start, Casie – Low Start, Ella – Medium Start).

Jim: High Start group

Jim's TPACK-CT trajectory started with a high numerical value of 3.3 in his programming assignment where he provided a sound explanation around the ways in which his lesson plan can support the development of student CT knowledge and skills. Jim's programming lesson focused on the integration of Scratch in Language Arts. Specifically, Jim's plan built on a lesson observed in his field placement where second grade students read about different forms of transportation. To extent the lesson, Jim planned to have students generate a story that uses two different forms of transportation (one modern form and one past form) and subsequently engage in peer reviews and iterative revisions. As a final step, the plan asked students to create an interactive digital story using Scratch. To achieve this goal, Jim proposed to cooperate with the school's technology teacher to introduce students to block-based programming. Jim's reason for incorporating Scratch (TK) into his Language Arts lesson was to help students make connections between the writing process and the "sequencing mindset that is needed to program things" (CT). Notably, Jim did not have prior programming experience when he enrolled in the course; however, he demonstrated positive attitudes towards Scratch with accurate understanding around the role of the tool in supporting the integration of CT with content and pedagogy (TPACK-CT).

In his second lesson plan, Jim focused on the use of a concept-mapping tool as a means of helping students visualize and represent two-step addition problems in mathematics. Similar to his programming lesson, Jim proposed to build up student knowledge of using concept mapping tools (TK) while simultaneously helping them develop conceptual understanding about addition and place values (i.e., content). Subsequently, he planned to provide students a math word problem and ask them to use the concept mapping tool to represent the solution (CT). Jim pointed out that using the concept mapping tool could benefit students' math conceptual understanding while supporting CT skills such as problem decomposition (TPACK-CT).

Jim was able to provide evidence of TPACK-CT in his programing and concept mapping lessons but not in his final case narrative where he implemented a stand-alone technology lesson with no connection to curricular content. Specifically, Jim's final case focused on the *Hour of Code* (see Table 1) as a way of helping students develop CT skills such as sequencing and algorithmic thinking.

Casie: Low Start group

Casie's programming lesson received a score of 1.25, which placed her in the Low Start group. In this lesson, Casie utilized Scratch games into a math lesson with the objective of helping students use place value understanding to whole number to the nearest 10 or 100. The use of programming, however, did not support the lesson's objectives. Casie simply assumed that playing Scratch games would deliver the math content. After playing the games, she planned to assess students using a worksheet. In this lesson, Casie demonstrated a misunderstanding of CT by asserting that it was the skill that "students would need to understand (when) they need to click on the right answer to complete the lesson". Reflecting on her experience, Casie noted that she found Scratch difficult and confusing.

In her second plan, Casie demonstrated a developing understanding of TPACK-CT. In this lesson, Casie utilized a concept mapping tool to help students understand a story. Her plan focused on reading the story of the *Three Little Pigs* to her students and subsequently completing a concept map focusing on the "who, what, when and how" of the story. In this lesson, concept mapping can help facilitate students' CT in the areas of problem decomposition (breaking up the story into smaller chunks) to facilitate understanding of the story. Although Casie provided a connection between the use of concept mapping within a content area, her pedagogical decisions on how to guide first graders use of concept mapping tools was relatively weak.

In her final assignment, Casie used an interactive whiteboard application to deliver a math lesson on fractions. Specifically, she presented 7 brownies and had students come up to the board to equally sort the brownies into four portions. This lesson could support students' CT knowledge and skills through problem decomposition and simulation with the use of technology. Casie, however, did not recognize these connections and rather indicated that the lesson supported students' CT because it allowed them to "be hands on with technology".

Ella: Medium Start group

Ella represented the group of pre-service teachers who scored between 2.2 to 2.9 in the programming lesson and steadily advanced throughout the duration of the course. Ella's programming lesson was designed to extent a math lesson observed in her field placement that focused on helping first graders practice addition and subtraction within 20. The plan engaged students with a series of progressively more difficult games in Scratch where they could practice addition and subtraction followed by completing an exit ticket aimed at checking understanding of relevant concepts. Finally, Ella's plan involved students creating their own addition and subtraction game in Scratch. Ella's lesson demonstrated a moderate understanding of TPACK-CT; although she incorporated programming into the lesson, she provided limited explanation of the technological and pedagogical decisions that would help support young students' math learning through programming. Unlike Casie, however, Ella reported a positive learning experience with Scratch programming.

In her concept mapping lesson, Ella aimed to help students find factor pairs for a whole number in the range from 1 to 100. To achieve this goal, she used a concept mapping tool to help students visually represent and organize math facts and solutions. Besides having whole class instruction, Ella also had students work on creating individual visual representations of pair factors. Ella's concept mapping lesson demonstrated a developing TPACK-CT since she was able to frame the use of a computing tool with content and pedagogical context. Besides helping students achieve the math learning goals, Ella's concept mapping lesson supported students' CT development in problem decomposition, simulation and algorithmic thinking.

Ella's final product, utilized various forms of technology in conjunction with a social studies concept related to the *Three Branches of Government*. In this lesson, however, the connection between technology (e.g., video and projector), content and pedagogy was relatively weak. The video contained probing questions that helped students break down the meaning of the text (e.g., problem decomposition), but Ella failed to conceptualize her content and pedagogical decisions associated with the use of the technology and how those may support CT.

Conclusion and implications

For pre-service teachers to successfully integrate CT in school curricula, they must develop a sound understanding of CT concepts, computing tools and practices. In this work we examined pre-service teachers' growing trajectories as they designed CT-infused lesson plans through their participation in an educational technology course offered in conjunction with field experience in authentic classrooms. Examination of pre-service teachers lesson designs indicated a growing understanding of TPACK-CT over time, despite differential starting points. Problem decomposition and algorithmic thinking were the most frequently cited CT constructs in participants' lesson designs. Overall, participants scored higher in the concept mapping lesson design; they illustrated a good understanding of TPACK-CT by using concept mapping tools to support both curricular objectives and the development of CT (e.g., problem decomposition, simulation, etc.). However, it is worth noting that this assignment was hypothetical – participants did not have to implement it.

In contrast, participants demonstrated a dip in the representation of TPACK-CT in practice. In this assignment, pre-service teachers encountered difficulties in either selecting appropriate computing tools or infusing CT concepts into the context of disciplinary content and pedagogy. Moreover, participants frequently failed to recognize other crucial CT concepts as they delivered their lessons including data analysis and automation. This finding is not surprising given participants' lack of classroom teaching experience. As a form of knowledge rooted in classroom practice TPACK-CT is influenced by pre-service teachers' understanding of pedagogy in relation to both content and computing tools (Mouza et al., 2017). For pre-service teachers to successfully build and TPACK-CT and apply it in practice, we must provide extensive opportunities to develop both their *theoretical* and *practical* knowledge (Gomez, Sherin, Griesdorn, & Finn, 2008).

References

Angeli, C., Voogt, J., Fluck, A., Webb, M., Cox, M., Malyn-Smith, J., & Zagami, J. (2016). A K-6 computational thinking curriculum framework: Implications for teacher knowledge. *Educational Technology & Society, 19*(3), 47–57.

Barr, V., and 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.

Code.org (2017). *Recommendations for states developing computer science teacher pathways*. Retrieved from https://code.org/files/TeacherPathwayRecommendations.pdf

CSTA & ISTE (2011). *Computational thinking. Teacher resources.* http://csta.acm.org/Curriculum/sub/CurrFiles/472.11CTTeacherResources_2ed-SP-vF.pdf

Gomez, L. M., Sherin, M., Griesdorn, J., & Finn, L. (2008). Creating social relationships: The role of technology in preservice teacher preparation. *Journal of Teacher Education, 59*(2), 117-131.

Harris, J., Grandgenett, N., & Hofer, M. (2010). Testing a TPACK-based technology integration assessment instrument. In C. D. Maddux, D. Gibson, & B. Dodge (Eds.), *Research highlights in technology and teacher education* (pp. 323-331). Chesapeake, VA: Society for Information Technology and Teacher Education (SITE).

International Society for Technology in Education (2016). National educational technology standards for students. Retrieved from http://www.iste.org

Mishra, P., & Koehler, M. (2006). Technological pedagogical content knowledge: A framework for teacher knowledge. *Teachers College Record, 108*(6), 1017-1054. https://doi.org/10.1111/j.1467-9620.2006.00684.x

Mouza, C., Yang, H., Pan, Y. C., 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).

Shulman, L. S. (1987). Knowledge and teaching: Foundations of the new reform. *Harvard Educational Review, 57*, 1-23. https://doi.org/10.17763/haer.57.1.j463w79r56455411

Wing, J.M. (2006). Computational thinking. *Communications of the ACM*.

Yadav, A., Gretter, S., Good, J., & McLean, T. (2017). Computational thinking in teacher education. In Rich, P. and Hodges, C. (Eds), *Emerging Research, Practice, and Policy on Computational Thinking* (pp. 205-220). Springer International Publishing.

Yadav, A., Hong, H., & Stephenson, C. (2016). Computational thinking for all: pedagogical approaches to embedding 21st century problem solving in K-12 classrooms. *TechTrends*, *60*(6), 565-568.

Yadav, A., Stephenson, C., & Hong, H. (2017). Computational thinking for teacher education. *Communications of the ACM, 80* (4), 55-62.