Teaching Computational Thinking in K-9: Tensions at the Intersection of Technology and Pedagogical Knowledge

Teresa Cerratto Pargman, Stockholm University, tessy@dsv.su.se
Matti Tedre, University of Eastern Finland, matti.tedre@uef.fi
Mattias Davidsson, Linnaeus University, mattias.davidsson@lnu.se
Marcelo Milrad, Linnaeus University, marcelo.milrad@lnu.se

Abstract: This work draws attention to the question of how in-service teachers learn to teach computational thinking and programming across subjects in K-9 education. Drawing on qualitative analyses of 298 reflective notes provided by 120 in-service teachers attending a professional development program, we pay attention to the following emergent themes: i)developing an understanding of the subject of computational thinking in K-9, ii)connecting programming with the teachers' own subject matters, iii)understanding the purpose of teaching computational thinking. These themes point at the importance of scaffolding in-service teachers to learn to program with educational materials, tools and programming environments but also with a sound and inspirational pedagogy that addresses the fundamental questions of: what, how and why programming and computational thinking should be part of the compulsory school curriculum?

Background

Whereas previous research in the Learning Sciences and CSCL have extensively focused on how children learn to program (Guzdial and Boulay, 2019; Kafai and Burke, 2014; Bers et al., 2014), and how to design programming environments aimed at scaffolding children's computer programming (Resnick et al., 2009), this work draws attention to how in-service teachers learn to teach programming and computational thinking in K-9. Since the publication of Wing's (2006) influential viewpoint article, the idea that computational thinking (CT) "is a fundamental skill for everyone, not just for computer scientists." (p.33) had an enormous impact worldwide. Many policy-makers, researchers and computer scientists found in Wing's essay a powerful reason for adding "computational thinking to every child's analytical ability" (p.33). However, learning programming and computational thinking is hard and teaching them as part of the school curriculum is troublesome (Webb et al. 2017).

Our work looks at the tensions that in service teachers experience when they attend a regional teacher professional development program aimed at capacity building in programming and computational thinking. Taking the concept called "pedagogical content knowledge" (Shulman, 1986) that addresses how the knowledge of the subject matter (i.e., CT core ideas) is connected with the knowledge of the pedagogy (i.e. how to teach CT core ideas), we argue that professional development programs need to pay attention not only to the teachers' development of programming and computational thinking skills but also to the pedagogical knowledge that can ensure in-service teachers are equipped to teach computer science in their own subject matters.

We develop this argument by presenting the main findings of a case study that addressed 200 in-service teachers in Sweden. Drawing on qualitative analyses of 298 teachers' reflective notes written during and after the program, our work contributes to identify three types of tensions the teachers experience during their professional development, and more in particular, in relation to: i)developing an understanding of the content and scope of CT in K-9, ii)connecting programming with the teacher's own school subject matters, iii)understanding the purpose of teaching CT.

Findings

Developing an understanding of the content and scope of computational thinking in K-9

While most of the teachers found the regional professional program "inspirational", "refreshing" and "helpful" and appreciated the variety of material, programming environments, tools and resources shared during the program, many teachers pointed to their anxiety and lack of confidence in relation to teaching a subject that they found "hard" and "complex". The following excerpt illustrates such a concern in the following way:

> Considering how everyday life looks like when it comes to digitization, there is no doubt that programming belongs in school and teaching. However, it is hard for me as a teacher to decide at what level I should teach, how long and how broad I should go into it. Further, I think it is not so nice to just oblige us to teach a subject that we haven't study in our professional education.

The excerpt points to tensions that emerge when teachers teach a subject they know very little about as it has not been part of their education. These tensions can be linked to the lack of "powerful knowledge" as formulated by Young (2013) in relation to the issue of entitlement and the management of the growth of teaching expertise (Winch, 2013).

Connecting programming with the teachers' own subject matters

A recurrent theme in the corpus is the teachers' interests in learning and implementing visual/block programming with the goal to program physical artifacts such as robots and/or engage in the making of simple electric/electronic constructions. The teachers mentioned they felt inspired by the playful nature of these tangible computer environments that definitely "catch the children's interests" and make of programming a fun school activity. However they also pointed to tensions that emerge between connecting the "exciting" and "fun" physical computing constructions with their own subject matter. The following excerpt gives us a glimpse about how the this issue.

> Next time I would like to know more about how to design a lesson in which we do programming, and how I should put programming into a thematic context within my own school subject so programming is not just "a fun thing" you do sometimes in the classroom.

The excerpt points to pedagogical issues pertaining to how balancing the excitement provoked by programming with the knowledge and skills sought to be developed in the teacher's subject matter.

Understanding the purpose of teaching computational thinking

From the analysis of the collected data it emerges that some teachers mentioned they wish they could articulate why computational thinking needs to be taught in K-9 education.

> There are many good ways to teach programming but what is important to me is to teach why we need to program.

Whereas the question of why teaching computational thinking in schools has been convincingly addressed earlier by Soloway (1983), we believe that it has to be further discussed in the context of how we scaffold K-9 teachers to develop *professional* and *meaningful* teaching competence in computer science.

References

Bers, M. U., Flannery, L., Kazakoff, E. R., & Sullivan, A. (2014). Computational thinking and tinkering: Exploration of an early childhood robotics curriculum. Computers & Education, 72, 145-157.

Kafai, Y. B. and Burke, Q. (2013). Computer programming goes back to school. Phi Delta Kappan, 95(1):61–65.

Guzdial, M. and du Boulay, B. (2019). The history of computing education research. In Fincher, S. A. and Robins, A. V., editors. The Cambridge Handbook of Computing Education Research, pages 11–39. Cambridge University Press, Cambridge, UK.

Resnick, M., Maloney, J., Monroy-Hernandez, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B., & Kafai, Y. (2009). Scratch: Programming for All. *Communications of the ACM*, vol. 52, no. 11, pp. 60-67.

Shulman, L. (1987). Knowledge and teaching: foundations of the new reform. *Harvard Educational Review, 57*(1), 1–22.

Young, M. (2013). Overcoming the crisis in curriculum theory: a knowledge based approach. *Journal of Curriculum Studies, 45*(2), 101–118.

Webb, M., Davis, N., Katz, Y, Bell, T.; Reynolds, N.; Chambers, D. and Syslo, M.(2017). Computer science in K-12 school curricula of the 2lst century: Why, what and when? In Educational and Information Technologies. 22: 445-468.

Winch, C. (2013). Curriculum design and epistemic ascent. *Journal of Philosophy of Education, 47*(1), 128–146.

Wing, J. (2006). Computational thinking. *Communications of the ACM, 49*(3), 33–36.