Challenges of preparing secondary subject pre-service teachers to integrate computational thinking into their teaching

Hannah Sabo

7 December 2021

What is helpful feedback?

- Presenting data from an in-progress paper
- Questions regarding the argumentation being made
- Insights into the data
- Further suggestions on closing the gap that we’ve identified
- Culture/interpretation of quotes

Motivation

- Computation has become increasingly relevant in 21st century society
- CT added to education standards around the world [1-2]
- Constraints of education – *integrated or standalone?*
  - Limited time in school day
- Training teachers to include CT
  - Likely something they did not get in their pre-university education
  - Must be robustly prepared to be able to teach computational thinking effectively
  - In-service and **pre-service teacher efforts**

[1] NGSS Lead States, Next Generation Science Standards: For States, By States. National Academies Press, 2013.
[2] Department for Education, “National curriculum in England: Computing programmes of study,” Gov Uk, 2013.

Report Recommendations – Teacher Education

(T1): determine the **scope** of **pre-** and in**-service science teacher** and science classroom computational needs at all levels, and **determine the quantity and duration of professional development necessary for teacher expertise to develop and to be maintained**;

(T3): **articulate and leverage theoretical frameworks for integrated computing education to guide the development of** classroom tools, software, content standards, assessments, and **expectations for teacher preparation** and ongoing teacher certification;

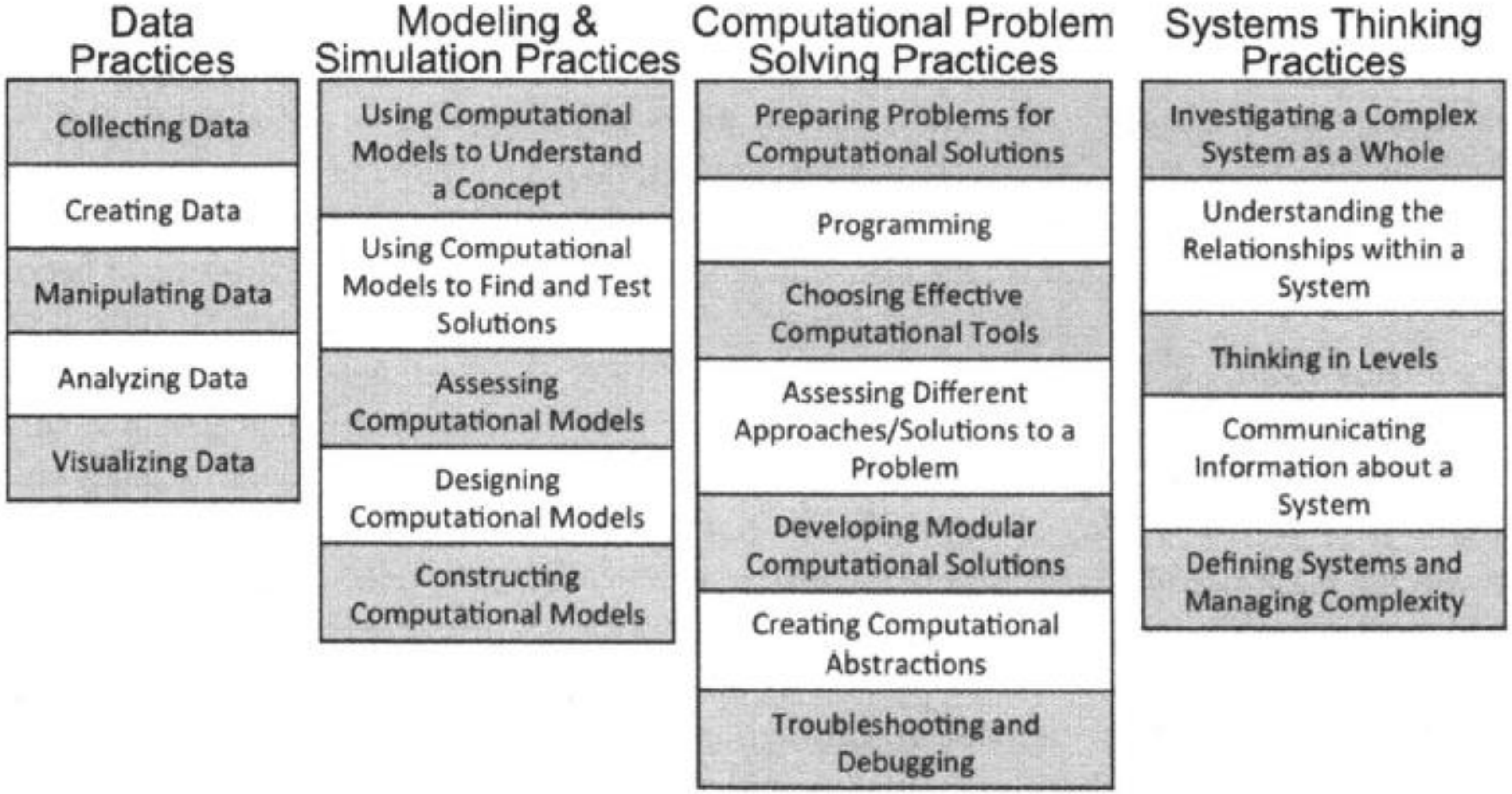
(T4): **understand whether and how teachers come to value computing as a component in science and mathematics education**

M. D. Caballero, K. Fisler, R. C. Hilborn, C. M. Romanowicz, and R. Vieyra, "Advancing Interdisciplinary Integration of Computational Thinking in Science," AAPT, College Park, MD, 978-1-931024-35–8, Jan. 2020.

Research Question

What challenges arise when preparing pre-service STEM teachers to integrate computational thinking into their teaching?

Defining Computational Thinking



D. Weintrop *et al.*, "Defining computational thinking for mathematics and science classrooms," *Journal of Science Education and Technology*, vol. 25, no. 1, pp. 127–147, 2016.

Background – How are teachers prepared to teach CT?

- In-service teachers
  - Professional development seminars [1]
- Pre-service teachers
  - Weeklong modules added to teaching methods courses [2]
- Creation of frameworks [3-5]
  - Designed to move computational thinking from abstract constructs to concrete skills/concepts/practices

[1] D. J. Ketelhut, K. Mills, E. Hestness, L. Cabrera, J. Plane, and J. R. McGinnis, “Teacher change following a professional development experience in integrating computational thinking into elementary science,” J. Sci. Educ. Technol., vol. 29, no. 1, pp. 174–188, 2020.
[2] A. Yadav, C. Mayfield, N. Zhou, S. Hambrusch, and J. T. Korb, “Computational thinking in elementary and secondary teacher education,” ACM Trans. Comput. Educ. TOCE, vol. 14, no. 1, pp. 1–16, 2014.
[3] V. Barr and C. Stephenson, “Bringing computational thinking to K-12: what is Involved and what is the role of the computer science education community?,” *Acm Inroads*, vol. 2, no. 1, pp. 48–54, 2011.
[4] D. Weintrop *et al.*, “Defining computational thinking for mathematics and science classrooms,” *J. Sci. Educ. Technol.*, vol. 25, no. 1, pp. 127–147, 2016.
[5] D. P. Weller, T. E. Bott, M. D. Caballero, and P. W. Irving, “Developing a learning goal framework for computational thinking in computationally integrated physics classrooms,” *ArXiv Prepr. ArXiv210507981*, 2021.

Background – What are the challenges of preparing teachers to teach CT?

- Hidden curriculum/lack of explicit goals; curricular time; range of instructor backgrounds; range of student backgrounds; variety of computational tools; inadequate textbooks; shortage of educational research; lack of community and support; space and scheduling constraints; hardware challenges; software installation. [1]
- Resources; Curricular constraints; need for professional support; need time; lack of familiarity; new way of asking students to think [2]

[1] AAPT Undergraduate Curriculum Task Force, "AAPT Undergraduate Curriculum Task Force - 2016 - AAPT Recommendations for Computational Physics in the Undergraduate Physics Curriculum," The American Association of Physics Teachers, Sep. 2016. [Online]. Available: https://www.aapt.org/Resources/upload/AAPT_UCTF_CompPhysReport_final_B.pdf

[2] E. Hestness *et al.*, "Computational thinking professional development for elementary science educators: Examining the design process," in *Society for Information Technology & Teacher Education International Conference*, 2018, pp. 1904–1912.

Secondary Pre-service teacher preparation at UiO

- 5-year program – resulting in Master’s degree
- Dual track
  - Subject courses – 2 subjects, one main, one secondary
  - Professional (teaching-related) courses
    - Pedagogy & didactics

Methods

- Informed by an internal survey given to the pre-service teachers (PSTs) in April 2021
  - Confidence and readiness to teach computational thinking
  - Learning about PSTs' experiences-related to computation and computational thinking
- Piloted the protocol with three students
- Initial reading of transcripts to identify themes
- Today, I am presenting one problem of practice: the pre-service teachers learned CT at a level above what they would be teaching. And that this does not immediately translate to their teaching

Camilla*, Lines 59-61

I think it would have been. **I would have liked to have sort of seen done, more programming with the level of maths that we're going to teach**. Because most of my programming has been done at a higher level in maths or physics. So, to have done it at a lower level would have been nice. **And to have done it practically because programming is something that you have to do to learn it.** Uh, so that's I think what I missed. What I feel like I've missed most... I can't really **use** the fact that I can find the eigenvalues or matrices in the uh high school level maths.

* All names are pseudonyms

Kristoffer, line 23

I believe that like the first year, the first semester, and the physics mechanical introduction course is the most relevant for programming that we are teaching in upper secondary school, the next years. Because they learn about using different, yeah. I guess you already know this, but **that meets the curriculum for the upper secondary school most, I think**. And there was several times that I felt that the um, **programming we did in the later physics courses were. It was nice to work with it, but that was way above what we’re supposed to teach.** Way. I think, so. But so, there's been continuous programming throughout the course.

Diana, line 38

So, from the 8th grade to the last year of high school, it's not the purpose that they're not going to be programming the whole thing, but they need to understand how the program works and how to use it. So, uh why we at university, we learn how to program everything from scratch. So right now for me, **it's unclear what the student is supposed to learn.** If they're going to do both or if they're just going to get a script that is finished and try to understand how it works.

Discussion

- Disconnect between the programming that they learned in their upper division subject courses and CT that they'll be teaching (C, N, D)
- Camilla's perspective – learning coding through doing;
  - Valued coding related to what they'd be teaching.
- Introductory courses more relevant to what they'll be teaching(Kristopher, Diana)
- Lack of clarity in what the standards expect of them (Diana)

Closing the gap – connecting computing experiences with teaching experiences

1. Opportunities to learn programming/computation in introductory-level content courses
   - Heavier stress of computation earlier. But students have constraints
2. Opportunities to learn computational thinking at the level that they'll be teaching
   - Likely in subject courses
3. Opportunities in methods/pedagogy courses to connect upper division computational thinking to the content that they'll be teaching.

Thank you!

- Any questions/comments for discussion?

Challenges of CT Integration	
Need resources for CT integration	• Integrating CT may be facilitated by access to resources (including funding) not readily available in elementary schools • Elementary teachers need curricular resources and practical examples of CT-infused learning activities, especially those that utilize available technology
Curricular constraints	• Elementary teachers must find ways to relate CT to the topics and themes they are already required to teach
Need professional support	• Elementary teachers need professional support to cultivate their understanding of what CT integration looks like in practice
Need time	• Time is limited for elementary science instruction (compared to reading or math); CT-infused science lessons require more time as students engage with challenging problems • Elementary teachers have limited time for planning
Lack of familiarity	• Colleagues and administrators may not be familiar with CT; elementary teachers seeking to integrate CT may have limited support or opportunities for collaboration
New way of asking students to think	• Students may become frustrated by challenging problems that require the use of CT • CT may require students to adopt a mindset that it's okay to fail

Table 1. Participant-Identified Challenges of CT Integration

From:

E. Hestness, D. J. Ketelhut, J. R. McGinnis, and J. Plane, "Professional knowledge building within an elementary teacher professional development experience on computational thinking in science education," *J. Technol. Teach. Educ.*, vol. 26, no. 3, pp. 411–435, 2018.