On Computational Thinking: A Perspective From Rural Saskatchewan

Episode 5: Teacher Education And Comfort With Computational Thinking Script

Stephen Hadden

Educational Technology and Design, University of Saskatchewan

ETAD 992: Project

Dr. Paula MacDowell

April 30, 2023

My interviewees, my research and my experience all point out something similar - time is short in an educator's day, and computational thinking for many teachers is kind of intimidating. How can we feel comfortable as educators to include coding, or other forms of computational thinking into our days? Let's explore. Thanks for joining me for this episode, this one is about Teacher Education and professional development. I'm Stephen Hadden.

Teacher education around computational thinking, or coding, whether pre-service or through professional development, may require more substance. I have taken a number of technology courses from undergraduate and graduate options at the University of Saskatchewan in my educational technology and design graduate study. But this is likely not the same experience for all. Here are a few quotes from articles that show gaps related to computational thinking and coding education for teachers. From the article "Coding in Primary Grades Boosts Children's Executive Functioning", "Yet, the schools in which coding has been regularly embedded in the STEM curriculum are still few, most teachers lack familiarity with coding resources" (Arfé et al., 2019). From the article, "Computational Thinking in Elementary and Secondary Teacher Education", "Most of the current efforts to educate teachers about CT have been limited to computer science teachers" (Yadav et al., 2014). From the article, "The Digital Skills Gap is an Issue for the Whole Tech Community":

> The provision and teaching support available for computer science.
>
> - no formal training or background (61%)
> - around 25% limited teacher knowledge and lack of digital skills. (Wood, 2022)

From the article, "The Code Centric Nature of Computational Thinking Education", "Only 19% of our sample reports programs for developing preservice and in-service teachers' understanding of CT or confidence in bringing CT to their classroom" (Kite et al., 2021). From the article,

“Facing the Challenges of 'Digital Competence'”, “More effort is needed to invest in teacher education and preparing student teachers’ digital competences” (Erstad et al., 2021).

Computational thinking, or coding, is not necessarily something that most elementary educators are entering the profession of education to teach. If it is not taught to them in university, then it makes sense that most educators, especially elementary educators, may be a little reluctant to embrace it. Furthermore, if mandates or documents from the government or school divisions express the need to explore computational thinking or coding more extensively, then the question becomes: Is this just something else to add to our teaching loads? It becomes a strain on precious time, rather than being given the exploration it may deserve. (Hennessey et al., 2017)

There are ways to help alleviate this potential strain though. The “Hiding in Plain Sight” article points out that identifying computational thinking concepts in existing curricula and practice can be a scaffold to encourage further incorporation of computational content in practice (Hennessey et al., 2017). Modules on computational thinking with pre-service teachers can shift thinking about the topic, increase comfort through familiarity, and provide opportunities to generalize principles - which means they are practising their own computational thinking skills (Yadav et al., 2014).

Making teachers aware that they already include computational thinking processes is a very important part to ensure that there can be further adoption. This stuff is in our current curricula, and it's already part of our practice. We may need to shift a few things to extend the integration and move toward new tools or practices, but first we can look for the positives in what we already do.

Next, we have to consider what forms of training we have for our educators, or as educators we need to see what is out there. Divisions and schools need to identify where their educators lack familiarity with the concepts and practices of computational thinking. We may have lots of people interested to explore further, but they may be intimidated, or unsure about where to start. I guess there are some questions school or division leadership need to consider: What professional development do we expect our teachers to attend? Do we have effective PD for this topic, or do we hope that teachers will find something interesting? There is plenty of good PD available through grassroots, national and international organizations - sometimes its a matter of knowing what will fit for our circumstances.

This leads to the next point. We need to make sure that the research about what works in the area of computational thinking gets to educators. Research on the quality and effectiveness of coding and computational thinking tools is relatively young (Arfé et al., 2019; Hennessey et al., 2017). Evidence like the effects on grade 1 executive functioning - is very interesting - I would never have found it, if I wasn't poking through other articles on the topic. But the authors point out - that they were able to find only one other study of a similar type (Arfé et al., 2019). Educators are happy to use tools that they know work well - but some one has to let them know (and likely pitch it really well).

Our educators have lots to do. If there are dedicated individuals appointed to seek out tools and curate those tools, that could save our teacher's time. Organizations like SaskCode, work to build resources and pick tools that will benefit Saskatchewan Educators. SaskCode has been able to expand their offerings with more funding and they provide PD and resources to divisions across the province (SaskCode, n.d.). They attended our recent local teacher's

association conference day - we had lots of fun getting robot mice to dance. I talk more about SaskCode in the resources episode, and extensively in the next interview episode.

Finally, to make computational thinking feel comfortable, or at least less intimidating, ask somebody technology-minded to suggest a coding tool or activity to explore with your students. Exploration and play are excellent ways to learn computational thinking. Let yourself and the children play with the tools for a period of time. You might not end up being the expert - one of your students may pick it up faster - but then let them explain it to you or use tutorials - there are many tools with excellent tutorials. Computational thinking has the potential to be about creativity and fun, it is a process to be explored. Build in some time to constructively explore the tools, it might lead to some class chaos, but if you are fine with that - this could be the best way to explore coding and computational thinking (Dougherty, 2012; Kite et al., 2021).

With a little knowledge and a little exposure, more teachers can find something that they really like doing with computational thinking and maybe even coding. And hey if you don't get coding, let someone come in who does know it, join the kids in the exploration and have a little fun with it yourself.

Having new tools and trying out new things is very exciting for me, that's why I love when I get my hands on some new robotic toy for class. But before we get to the tools, we have one final interview. Dean Elliott was a science curriculum consultant at the ministry for many years. He recently retired but is now working with SaskCode as the provincial outreach coordinator. Dean has a great deal of insight into the development of computer science and coding instruction in the province and has been involved with the growth of SaskCode. Then please join me for one final episode where I get to talk about some of my current favourite tools and strategies. Thanks for listening, I'm Stephen Hadden.

References

Arfé, B., Vardanega, T., Montuori, C., & Lavanga, M. (2019). Coding in Primary Grades Boosts Children's Executive Functions. *Frontiers in Psychology*, *10*, 2713. https://doi.org/10.3389/fpsyg.2019.02713

Dougherty, D. (2012). The Maker Movement. *Innovations: Technology, Governance, Globalization*, *7*(3), 11–14. https://doi.org/10.1162/INOV_a_00135

Erstad, O., Kjällander, S., & Järvelä, S. (2021). Facing the challenges of 'digital competence.' *Nordic Journal of Digital Literacy*, *16*(2), 77–87. https://doi.org/10.18261/issn.1891-943x-2021-02-04

Hennessey, E. J. V., Mueller, J., Beckett, D., & Fisher, P. A. (2017). Hiding in Plain Sight: Identifying Computational Thinking in the Ontario Elementary School Curriculum. *Journal of Curriculum and Teaching*, *6*(1), 79. https://doi.org/10.5430/jct.v6n1p79

Kite, V., Park, S., & Wiebe, E. (2021). The Code-Centric Nature of Computational Thinking Education: A Review of Trends and Issues in Computational Thinking Education Research. *SAGE Open*, *11*(2), 215824402110164. https://doi.org/10.1177/21582440211016418

SaskCode. (n.d.). *SaskCode*. SaskCode. Retrieved January 1, 2021, from https://www.saskcode.ca/pd-events

Wood, M. (2022). *The digital skills gap is an issue for the whole tech community*. Eandt.Theiet.Org. https://eandt.theiet.org/content/articles/2022/10/the-digital-skills-gap-is-an-issue-for-the-whole-tech-community/

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*, *14*(1), 1–16. https://doi.org/10.1145/2576872