Nordic Approaches to Computational Thinking in Teaching and Learning

Host names: Johan Lundin (University of Gothenburg), Renate Andersen (Oslo Metropolitan University), Teemu Leinonen, Nina Bonderup Dohn (University of Southern Denmark), Håkon Swensen, Louise Mifsud (Oslo Metropolitan University), Anders Morch (University of Oslo), Anders Kluge (University of Oslo), Stig Børsen Hansen (University of Southern Denmark), Daniel Spikol, Jussi Mikkonen (University of Southern Denmark), David Cuartielles, Rocio Chongtay (University of Southern Denmark)

Abstract

- Nordic countries have recently implemented changes in the curriculum introducing computational thinking integrated in Mathematics, Natural Science, Arts and Crafts and Music. This poses challenges as well as opportunities, in particular for teacher education, as it is responsible for educating teachers who can teach computational thinking skills within these subjects.

Round table description

In this round table, we focus on critically examining the notion of computational thinking (e.g., Denning & Tedre, 2019) in the context of teaching and learning. For example, teacher education has the mandate to educate teachers who can teach pupils programming skills according to curricular demands and programming related to computational thinking. Given the current capacity of training teachers in learning and teaching programming in the Nordic countries, we are dependent on teachers' engaging in networked learning in professional communities, including teacher communities of practice and online access to best practice by examples.

Another example is to examine to what extent the practice of programming (learning and doing it) can develop pupils' computational thinking skills and vice versa if knowledge of computational thinking can improve one's skills in programming. We consider these concepts as in tandem, where one necessitates the other. Another approach will be to question whether programming would lead to improved skills and competency in computational thinking. It is also relevant to question the concept of computational thinking in light of what the future citizens would need in order to handle the increased digitalization of society, organizations and leisure life. In addition, we ask the question whether it is a good idea to teach and learn three things at the same time: Computational thinking, programming, and subject matter (math, natural science, etc.)

The roundtable addresses the challenges and opportunities that the introduction of computational thinking in compulsory schooling leads to, as seen from a networked learning (NL) perspective. We understand networked learning along the lines of the second approach in the overview provided by Dohn et al (2018) of Networked Learning Conference perspectives, focusing on *learning networks*: "What makes learning 'networked"' is the connection to and engagement with other people across different social positions inside and outside of a given

institution. The network is supportive of a person's learning through the access it provides to other people's ideas and ways of participating in practice" (p. 204). Learning networks need not be mediated by ICT within this approach but may also refer to co-located people (small group participation) who engage in a range of hybrid virtual-physical learning activities (Dohn, Sime, Cranmer, Ryberg, & De Laat, 2018).

We suggest discussing the following questions in a plenum round table at the conference, which we believe is of interest to the general audience:

- Is computational thinking the right concept for promoting digital citizenship and empowerment?
- What are the arguments for introducing computational thinking and/or programming in school (all Nordic countries are in a similar position when it comes to integrating programming in specific subject areas such as math and natural science, but also arts & crafts and music in some countries)?
- The relationship between computational thinking and programming is well established in computer science, but will computational thinking for computer science integrated into other subjects serve a useful purpose, or will it take focus away from the subject matter?
- How can computational thinking support collaborative and networked learning processes?

Goals of the round table - collective outcome

- Input from the audience to what should be a working definition of "Computational Thinking" of practical relevance for teacher education, focusing on teachers' communities of practice as "learning networks". We aim to achieve this by combining a top-down and a bottom-up process. Top down by looking at how computational thinking is operationalized in the Nordic curricular/policy documents (a preparatory activity by the panelists), and bottom up by the ideas emerging in the roundtable discussion.
- Broadening the scope of participation to other schools / research groups in the field.
- Presentation of annotated exemplars of what "Computational Thinking" cases could be both as a subject in its own right and as applied to or integrated in other subjects.
- Identifying relevant interdisciplinary literature.
- A critique of current, inadequate/outdated definitions of CT? (It seems the original idea is interesting but that it has been simplified or "watered" out in later / contemporary interpretations. Should we go back to sources or start from the current best definition?)
- Practical examples (of using CT) in learning and in teaching programming courses such as 1) studies of teaching it and 2) teacher cases. For example, we could be collecting a broad scope of practices that might create a good pool of material for directing work towards best practices, and driving the development of new methods.
- Making distinctions and types of practice with ICT (e.g. to distinguish between programming in hardware-oriented approaches (Micro: bit/Arduino; maker spaces) vs programming for non-specific hardware (Scratch/Python on screens only), distinguishing different ways of invoking a program (event-based, automata-based, flow-based, etc.)

- Distinguishing programming environments from e.g. domain-oriented design environments; where in the latter programming is introduced to solve breakdowns (e.g. programming in Minecraft to enhance game experience and make tedious tasks simpler)
- Collaboration/Nordic unique perspectives: Computational participation and collaborative aspects is part of the Nordic tradition of programming and systems development that started back in the 1970-80s with Participatory Design projects (Nygaard, 1986) and treating programming as theory building (Naur, 1985).
- How do we differ from other popular approaches/perspectives that also advocate participation/collaboration (e.g., Kafai, 2016)
- How computational thinking make sense in educational policy documents and national educational plans such as curricula for specific subjects with core concepts and learning goals?

How to engage the participants in the discussion

- We ask panelists and audience to bring (a) case(s) for discussion, good, bad, and ugly examples.
- Panelist(s) will show examples of empirical research (results and data) with the aim to present in non-technical language to motivate audience participation
- Demonstration prototypes tested out in one or more educational institutions
- Engage the audience in using computational tools (e.g. Arduino or Minecraft).
- A short design experiment for anyone to understand the complexity of the term.

References

Denning, P. J., & Tedre, M. (2019). *Computational Thinking*. Cambridge, MA: MIT Press.

Dohn, N. B., Sime, J.-A., Cranmer, S., Ryberg, T., & De Laat, M. (2018). Reflections and Challenges in Networked Learning. In N. B. Dohn, S. Cranmer, J.-A. Sime, T. Ryberg, & M. De Laat (Eds.), *Networked Learning: Reflections and Challenges* (pp. 187-212). Cham: Springer.

Kafai, Y.B. (2016). From Computational Thinking to Computational Participation in K-12 Education. *Communications of the ACM*, Vol. 59 No. 8, pp. 26-27.

Naur, P. (1985). Programming as Theory Building. *Microprocessing and Microprogramming*, 15(5), 253-261.

Nygaard, K. (1986). Program Development as a Social Activity. In H.-J. Kugler (Ed.), *Information Processing 86* (pp. 189-198). Amsterdam: North Holland.