A qualitative study of the enablers and inhibitors encountered by primary school teachers preparing for the implementation of Digital Technologies in New South Wales

Dr Daniel Hickmott, University of Sydney
Associate Professor Elena Prieto-Rodriguez, University of Newcastle

About me

Completed my PhD (Education) in September 2020

- Supervised by A/Prof Elena Prieto-Rodriguez and Prof John Fischetti

Thesis titled: *A study of a professional learning program for primary school teachers implementing the New South Wales science and technology syllabus*

- hckmd.com/thesis
- Focus of this presentation is on inhibitors and enablers, identified through qualitative methods

Since early 2020, an Associate Lecturer, mainly teaching in units of the Master of Education (Digital Technologies) at the University of Sydney

Australian Curriculum: Digital Technologies (AC:DT)

F-10 subject, within the Technologies learning area

"empowers students to shape change by influencing how contemporary and emerging information systems and practices are applied to meet current and future needs."

teacher.codes/ACDT-rationale

Reflects trends internationally, introducing coding and computational thinking from early years of schooling, e.g., New Zealand and UK (Webb et. al, 2017)

Australian Curriculum: Digital Technologies (AC:DT)

teacher.codes/ACDT-structure

Two strands:

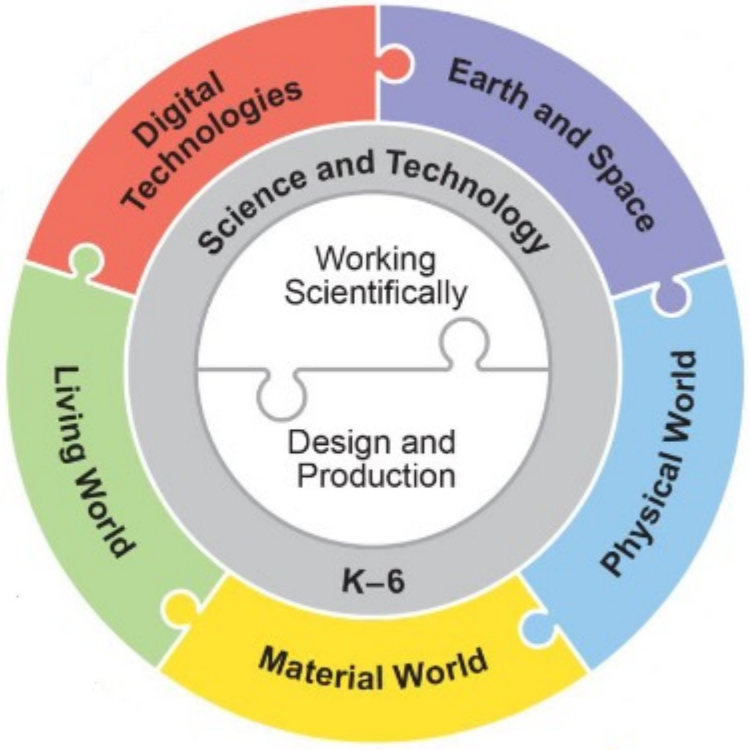
- Knowledge and understanding
- Processes and production skills

Underpinned by 10 key concepts, including **algorithms** and **implementation** (mostly equivalent to *programming* or *coding*)

Digital Technologies in New South Wales syllabuses (K-8)

Structure of Australian Curriculum (F-10)

NSW Science and Technology K-6



Challenges for teachers include:

Lack of instructional time and packed curricula (Israel, Pearson, Tapia, Wherfel, & Reese, 2015)

Low levels of confidence about teaching coding (Bower et al, 2017)

Lack of access to appropriate resources, both technological and educational (Kadirhan, Gül, & Battal, 2018)

Proposed strategies and solutions

Encourage integration, e.g., teaching coding alongside Maths

- Save instructional time? (Israel et al., 2015)
- Enhance the teaching of abstract topics (Benton, Saunders, Kalas, Hoyles, & Noss, 2018)

Professional learning and development of resources, funded by government, private companies and not-for-profit organisations

- MOOCs
- Workshops
- Interactive courses

Coding & STEM 4 Schools & Coding in Stage 3

Projects I worked on at the University of Newcastle 2013-2019

Mainly face-to-face workshops, run over 2 days on campus

- Rationale that this mode was preferred by teachers with low confidence about teaching concepts (Sentance & Humphreys, 2015)

We applied we learned from the workshops we ran, literature reviews and collaborations with others to the Coding in Stage 3 program, which ran in 2018 over 10 weeks.

Coding in Stage 3 (CS3)

Open to those teaching Stage 3 classes (Years 5 & 6)

Teachers learned coding over 10 weeks, with a focus on using Scratch, in weekly 2-hour afternoon school workshop sessions

Two streams:

- ScratchMaths
- Coding & STEAM

Pre- and post- surveys (n=42) then interviews (n=15) a few months after

Research Questions

First three questions related to impact of CS3 program, with sub-research questions comparing outcomes of different streams:

- Understanding of computational concepts
- Confidence in ability to teach computational concepts
- Plans for integrating coding in different subjects

RQ4. What enablers and inhibitors do primary school teachers encounter when implementing coding and computational thinking lessons after completing a sustained professional learning program?

Data collection and analysis

The interviews' durations ranged from 20 minutes to an hour

Inductive thematic analysis (Braun & Clarke, 2006) was applied

- Enabler: Teachers found links between coding and subject content easily
- Inhibitor: Teachers were challenged by an already 'packed' curriculum

16 enablers, 18 inhibitors identified

These were then combined into 9 categories that contained related inhibitors and enablers

Identified Categories of Enablers & Inhibitors

Curriculum	**Time**
Teachers	**Students**
Integration	**External and Extracurricular**
Technology (Software & Hardware)	**Leaders**
Peers	

Conclusion: Teachers were lacking confidence

Teachers' self-efficacy was very low before they completed the program (evident in quantitative results and discussion in interviews)

"If someone had said to me prior [to the course], 'Okay, we want you to teach coding', I would have probably just hid in the corner." (Letitia, Female Teacher, Coding & STEAM)

Important to note: teachers completed the program in their own time

Conclusion: Teachers were lacking confidence

Teachers' self-efficacy significantly increased and with large effect as a result of participating in the program

"I started knowing nothing about Scratch... now I feel confident that I have mastered the basics and would be able to take students on a similar learning journey to my own." (Letitia, Female Teacher, Coding & STEAM)

A 20 hour (low-cost) sustained program substantially improved teachers' self-efficacy

Conclusion: Integration may not save time

Teachers discussed choosing to teach coding in 'standalone' lessons, rather than integrated

"I've tended to teach coding and computational thinking as a kind of a standalone group of lessons, just while they understand the basics of how to code, and what you can do with coding, rather than lots of integration into other areas." (Natasha, Female Teacher, ScratchMaths)

Conclusion: Integration may not save time

Teaching coding is often a responsibility of teachers not in classroom teacher roles

"I do it in isolation to everybody else, basically, so I just grab the indicators from the Science syllabus now and focus on those." (Gabrielle, Female Teacher, Coding & STEAM)

Integrating coding into other subjects may not save time for teachers

Conclusion: Integration must be meaningful

Majority of teachers saw opportunities for integrating coding and Maths

"It's tied in very well to our maths, as well, because we just recently were doing coordinates and things like that, and the cartesian plane. So it's been excellent for that, 'This is X axis and Y axis'" (Letitia, Female Teacher, Coding & STEAM)

Conclusion: Integration must be meaningful

Teachers were concerned about 'surface-level' links between coding and other subjects.

"Like with the persuasive writing one I’m not assessing the actual persuasive writing in that because you wouldn’t see enough in a Scratch to do that.... I feel like I’ve used some of it but it’s all very surface level." (Belinda, Female Teacher, Coding & STEAM)

Need for further work and development of resources

Conclusion: Schools need shared vision

Some of the teachers were the only teacher in their school implementing coding

"There's 19 teachers in the school and no-one else does anything." (Gillian, Female Teacher, Coding & STEAM)

Some teachers were drivers of technology in their school, others were often relief from face-to-face (RFF) teachers or teacher-librarians

Conclusion: Schools need shared vision

Teachers discussed support from their school's executive and collaboration with colleagues

"Our principal gave us some time in lieu, so it really just supported the teachers to put the time and effort in to the program." (Alison, Female Teacher, ScratchMaths)

Individual teachers can be up-skilled for coding but there needs to be collaboration within schools

Would you like to know more?

Copy of the slides: hckmd.com/aare2021

Resources from Coding & STEM 4 Schools project: cs4s.github.io

My thesis: hckmd.com/thesis

Draft paper on learning design of Master of Education (Digital Technologies) units - happy to share via email

References

Benton, L., Saunders, P., Kalas, I., Hoyles, C., & Noss, R. (2018). Designing for learning mathematics through programming: A case study of pupils engaging with place value. *International journal of child-computer interaction*, *16*, 68-76.

Bower, M., Wood, L. N., Lai, J. W., Highfield, K., Veal, J., Howe, C., ... & Mason, R. (2017). Improving the computational thinking pedagogical capabilities of school teachers. *Australian Journal of Teacher Education*, *42*(3), 53-72.

Braun, V., & Clarke, V. (2006). Using thematic analysis in psychology. *Qualitative research in psychology*, *3*(2), 77-101.

Kadirhan, Z., Gül, A., & Battal, A. (2018). Self-efficacy to teach coding in K-12 education. In *Self-Efficacy in Instructional Technology Contexts* (pp. 205-226). Springer, Cham.

References

Israel, M., Pearson, J. N., Tapia, T., Wherfel, Q. M., & Reese, G. (2015). Supporting all learners in school-wide computational thinking: A cross-case qualitative analysis. *Computers & Education*, *82*, 263-279.

Sentance, S., & Humphreys, S. (2015, September). Online vs face-to-face engagement of computing teachers for their professional development needs. In *International Conference on Informatics in Schools: Situation, Evolution, and Perspectives* (pp. 69-81). Springer, Cham.

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