Primary school technology curriculum's computational thinking

KATE RHODES AND MELINDA DIXON

KEY POINTS

- Computational thinking is the thinking that takes place by a human when problem solving in computer science.
- *The New Zealand Curriculum* indicates that computational thinking can be taught without a device until Year 5.
- Unplugging (without a device) computational thinking can be beneficial to both students and teachers.
- Providing contexts for computational thinking that reflect the cultures and worldviews of students is important.
- Assigning student roles that emulate those in digital technology-related industries is effective.

https://doi.org/10.18296/set.1539

This article unpacks the computational thinking for digital technologies (CTDT) technological area of the New Zealand primary curriculum that became mandatory in 2020. It explores computational thinking (CT), its place in the New Zealand primary curriculum, and elements kaiako can utilise to help implement the teaching of CT. The article considers teaching CTDT through unplugged (without a device) activities, pedagogy, and contexts that reflect the worldviews of ākonga, with associated activities suggested. Furthermore, this article draws on several research projects where teachers implemented CTDT skills in their classrooms and explored unplugged concepts. It is envisioned that this article will change teachers' perceptions of teaching CTDT; from an unattainable set of specialist (digital) skills, to a set of day-to-day problem-solving techniques (applied to both unplugged and later-on-device contexts), and will support educators in understanding and implementing CTDT.

In today's world, digital technology (DT) saturates many aspects of our lives. In education, devices are utilised for administration purposes and as teaching tools. In our digital technology-laden world, it has become ever more imperative that digital understanding is developed in education, to enable ākonga to be skilled and effective users and creators of digital tools. The DT sector is considered to be a rapidly growing component of New Zealand's economy with many future career paths for young people of Aotearoa (Digital Skills Forum, 2021, p. 7). In 2017, recommendations were made by the Digital Skills Forum "to make sure every child is exposed to digital technology pathways" and "actively encourage a more diverse group of Kiwis into digital technology" (2021, p. 7). However, between 2017 and 2021, New Zealand experienced a decrease in DT in education and in diversity in the digital workforce (Digital Skills Forum, 2021). With a need for digitally fluent graduates, we have seen the development of primary (ages 5–12) digital curricula in New Zealand, known as digital technology and hangarau matihiko, with new DT areas added to the Technology learning area of the curriculum in late 2017 (Te Kete Ipurangi [TKI], n.d.-a). Specifically, digital technology and hangarau matihiko encompasses CT skills.

What is computational thinking?

The word *computational* suggests the use of a device such as a computer. However, computational thinking (CT) is about humans' mental problem solving and is regarded as a cluster of problem-solving skills that connect to principles of computer science (Curzon et al., 2009). Wing (2006) argues that CT is a fundamental skill for people in day-to-day life, just like literacy and numeracy. CT underpins computer science: however, CT is the *"thinking"* undertaken by a human to solve real-life problems both before and during our time on a device. Wing defines CT as:

> The thought processes involved in formulating a problem and expressing its solution(s) in such a way that a computer—human or machine—can effectively carry out. Informally, computational thinking describes the mental activity in formulating a problem to admit a computational solution. The solution can be carried out by a human or machine. This latter point is important. First, humans compute. Second, people can learn computational thinking without a machine. (2017, p. 8)

Within CT problem-solving processes, Astrachan (2009) suggests CT involves breaking down problems and solving parts one by one and/or modelling the problem clearly. Given this and Wing's (2017) definition and, importantly for educators, the most prevalent part of CT is the *thinking completed by humans*.

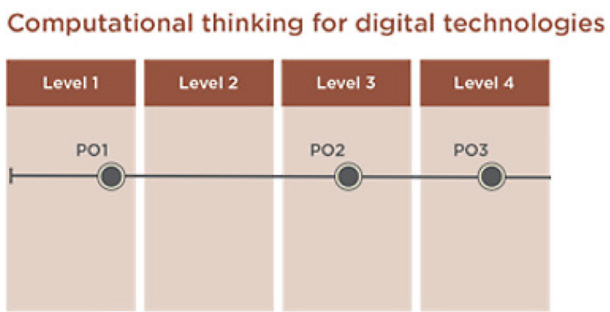
Coding (referred to as programming in *The New Zealand Curriculum*) is closely related to CT, with elements of CT and coding intricately interwoven. However, CT and coding are not the same, although developing one will theoretically develop skills in the other (Akiba, 2022; International Society for Technology in Education, 2022; Sun et al., 2021). Where CT is the thinking by humans, coding is the procedure of developing step-by-step instructions that a digital device can understand to programme its functions (McLennan, 2017). Given the difference between CT and coding, teaching CT effectively requires teachers understanding CT and coding and how they are linked.

Computational thinking in the New Zealand primary curriculum

Two new digitally focused technological areas—computational thinking for digital technologies (CTDT) and designing and developing digital outcomes—became a mandatory part of the New Zealand primary curriculum Technology learning area in 2020. The new areas aim to develop ākonga who are critical, creative, and reflective inventors and producers of digital concepts (TKI, 2018). When introduced, it was proposed that New Zealand teachers and schools would find adopting and implementing the new areas challenging (Education Review Office [ERO], 2019). This is because they involve expertise that is often outside the skill sets and experiences of many New Zealand primary teachers' current understanding of digital technologies (Calder & Rhodes, 2021; Crow et al., 2019; ERO, 2019; Rhodes & Calder, 2022). Although the two areas are interlinked and mutually influencing, our focus is solely on CT concepts for this article. In this article, we propose that shifting teachers' perceptions of CT from an unattainable set of specialist (digital) skills to a set of day-to-day problem-solving techniques (applied to both on-device and unplugged contexts) will help educators effectively understand and implement the CTDT area of the technology curriculum.

New Zealand primary school curricula outline three CTDT progress outcomes (POs) (see Figure 1). POs signal progression and specificity in content. POs assist teachers to understand and teach new content knowledge related to digital technologies. They are not to be used as a set of skills to be "ticked off" as they are learnt, rather they are designed to inform quality technology practice within authentic contexts (Fox-Turnbull et al., 2021).

FIGURE 1. PROGRESS OUTCOMES OF CTDT ALIGNMENT TO THE NEW ZEALAND CURRICULUM LEVELS



Note: From TKI, n.d.-b, para. 1.

The fundamental element of these CTDT is creating a set of instructions (program) for solving a problem. At junior levels, ākonga might create a sequence of steps or set of instructions without a computer (unplugged), while seniors might create sequenced instructions using **both** unplugged and computerised environments such as Scratch. Various lists of CT skill sets appear in literature, such as Butler and Leahy (2021). These align with the CTDT and are identified in Table 1.

TABLE 1. LIST AND DEFINITION OF COMPUTATIONAL THINKING SKILLS

CT skill	Explanation
Abstraction	Identify the problem that needs to be solved. Are there unnecessary details that can be removed? What can you do to make the process simpler?
Algorithmic Thinking	Create precise, unambiguous, step-by-step instructions for end users to follow. Consider inputs, outputs, sequence, and repetition of parts of the sequence (loop).
Decomposition	Break down a big problem into smaller, more manageable chunks.
Generalisation	Apply approaches and solutions from previous problem-solving scenarios to new problem contexts.
Debugging	Identify and fix errors as algorithms are followed.
Iteration	Repeat and review algorithms to refine solutions. Was there a more efficient way to solve this problem?

Note: From Butler and Leahy (2021); Te Kete Ipurangi (2018).

Importantly, in *The New Zealand Curriculum*, CTDT only has one PO below level 3 and does not outline the need for a computer for ākonga in this PO. Therefore, up until level 3 in New Zealand primary school CTDT, there is no requirement to use a computer. Instead, the use of unplugged activities is deemed suitable.

International and local research on unplugged CTDT

It is widely agreed that the end goal of teaching CT is to apply these specific thinking skills to computerised contexts. Nonetheless, studies suggest that unplugged lessons are just as effective, if not more, at promoting CT than those that are carried out using on-device contexts (see, for example, Hermans & Aivaloglou, 2017; Metin, 2022; Wohl et al., 2015). Experience with unplugged CT tasks at lower year levels has also been found to make learning more relatable for young children, develop student self-efficacy, and therefore make CT skills become more accessible (Relkin & Strawhacker, 2021). Developing this foundation of CT awareness, where students can understand CT skills and know when to apply these, has also been found to promote higher levels of achievement in computer science (CS) in secondary schooling (Relkin & Strawhacker, 2021). In fact, Fletcher and Lu (2009) suggest that to successfully develop CS in later schooling years, "efforts must be made to lay the foundations of CT long before students experience their first programming language" (p. 23). The benefits of teaching unplugged coding at earlier stages of the curriculum have also been found to develop teacher confidence and content knowledge while teaching DT (Bell & Vahrenhold, 2018; Duncan, 2018; Rhodes, 2020).

Our teachers' experiences unplugging CTDT

The first author's, Kate Rhodes', recent studies have explored teachers' implementation of CTDT (Calder & Rhodes, 2021; Rhodes, 2020). Given the previously outlined differences between CT and coding, Rhodes (2020) found all teachers in the study initially considered that CTDT required ākonga to complete complex, on-device coding tasks throughout primary school. When unplugged activities were unpacked and identified, including in isolation or in support of on-device activities, teachers felt the CTDT aspects were more accessible and simpler than first imagined (Rhodes, 2020).

Data from our research identified teachers' initial misconception that all CTDT needed to be completed on devices at all ages,[1] yet the research demonstrated that unplugged CTDT is fundamental and that CTDT often incorporated elements of other subjects such as maths, language instructions, and storytelling (Rhodes, 2020). Several teachers articulated that teaching through unplugged concepts simplified CTDT. They recognised the ability to integrate CTDT into other areas of the curriculum such as literacy, including report writing, procedure writing, and oral language; and numeracy, including direction, geometry, and other strand and number concepts. Teachers found utilising the unplugged concepts beneficial for their pedagogy and ākonga understanding. Another of our projects, undertaken with Year 5 and Year 6 students, examined teacher practice with coding through the use, evaluation, and adaption of University College London's ScratchMaths resources (Calder & Rhodes, 2021). In this study, the participating teachers found adapting a dual approach of unplugged and on-device teaching was beneficial to coding concepts, helping students to consolidate their learning in CTDT and across other curriculum areas. Teachers were grateful when they realised that CTDT concepts do not have to be taught on devices and felt this simplified teaching and learning of CTDT. In both studies, teachers gained confidence in teaching CTDT once they had unpacked CTDT and explored how unplugged elements could be developed and embedded into multiple curriculum areas. Our first study (Rhodes, 2020) also found teachers' confidence noticeably changed throughout the project, once a broader understanding of the unplugged activities developed, as shown in Table 2.

TABLE 2. COMPARATIVE COMMENTS AND REFLECTIONS BEGINNING TO END OF THE RESEARCH

	Beginning comments/reflections	**End comments/reflections**
Teacher 1	... there are a lot of people that feel apprehensive because they think it's [CTDT] so broad ...	CTDT is not as hard as I imagined.
Teacher 2	This is where I get confused about what CTDT does and what it doesn't ...	CTDT is not that difficult it's not this complete overhaul.
Teacher 3	I have no idea how to do that stuff! [CTDT]	I feel more confident implementing the CTDT after these lessons. It's less daunting than I originally thought.

Note: From Rhodes (2020).

Teaching to promote CTDT

Given the findings highlighted above, this current article provides teacher readers with several activities they can draw from to develop unplugged activities in their classrooms. When designing specific activities for their classrooms, kaiako should consider wider pedagogical strategies such as:

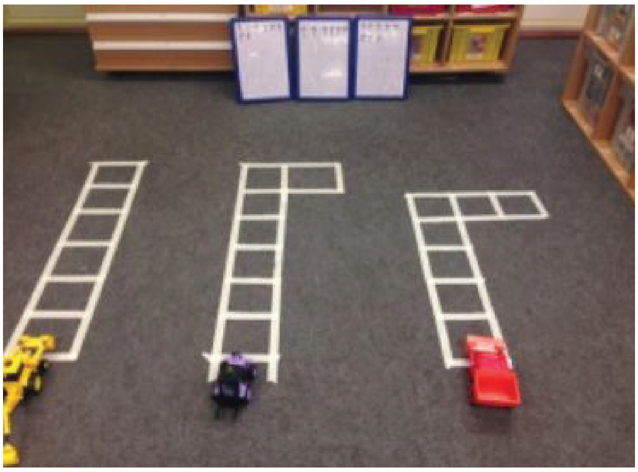
- exposing learners to multiple scenarios
- utilising contexts that reflect the worldviews of learners and their whānau
- assigning student roles in CTDT learning experiences that replicate those in the programming industry.

Examples of these strategies are woven through the activities and narrative presented below. Also, when teaching unplugged activities, teachers are encouraged to utilise the above list and definition of computational thinking skills identified in Table 1. Unpacking these with students can be a valuable activity to undertake at any year level.

CT activities with assigned roles

The following example shows how teachers can explicitly teach the CT skills of algorithmic thinking and debugging at lower levels of the curriculum through a simple unplugged task. In this activity, the teacher creates several grids (roads) on the floor using sticky tape, chalk, or pen and paper. Ākonga are then provided with three programs that they need to attribute to each grid. To match these programs, students are encouraged to move toy cars through the grid following the step-by-step instructions in each program. Next, learners work to create their own grids and programs for others to match. These might be more complex grids with obstacles and could also include self- and peer assessment. While doing this, ākonga can articulate step-by-step instructions to encourage oral literacy (forward, turn right (or turn), forward, forward), mathematical directional language, and collaboration. Finally, to promote an understanding of debugging, teachers can provide a new step-by-step set of instructions (program) that contain errors. Students need to identify and debug (correct) these. The utilisation of language such as algorithms, program, and debugging etc. is encouraged.

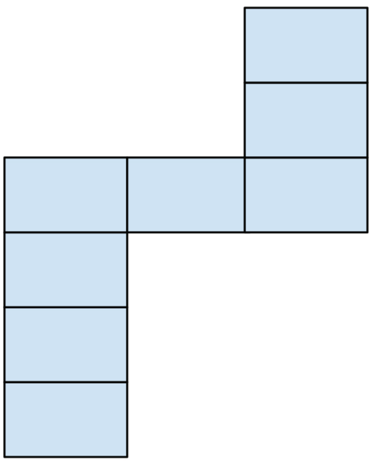
FIGURE 2. CRACK THE PROGRAM UNPLUGGED



Program 1: ↑ ↑ ↑ ↑ ↑ ↑
Program 2: ↑ ↑ ↑ ↑ ↑ ↑ → ↑
Program 3: ↑ ↑ ↑ ↑ ↑ ↑ → ↑↑

Note: From Withersey (2015).

FIGURE 3. GRID EXAMPLE



(Error) Program: ↑ ↑ ↑ ↑ ↑ → ↑↑ → ↑↑

To encourage CT skills more explicitly, teachers can assign student roles that align with individual CT skills that replicate those from the DT industry. The ability to work collaboratively in a team is considered important for success across a range of industries including DT (Ellington & Leonard, 2021; Hunt & Bers, 2021). Programmers deal with complex problems incorporating various interrelated steps that require teams of people to solve. These problems are often broken into several sub-problems (decomposition) that are then investigated and solved individually before being combined as a final outcome (Beecher, 2017). In the DT industry, programmers often work in teams where, in elementary terms, algorithms are created, tested, and debugged. Assigning students roles that replicate this process such as Programmer, Tester, and Debugger highlights the importance of collaboration, emphasises the need for CT competencies, and shows how CT skills complement each other to solve problems in CS. Other roles such as Evaluator where a student considers the efficiency of solutions or other potential pathways could also be created by teachers (Curzon et al., 2014). In the classroom, this approach provides a guide for working collaboratively towards a common outcome and prevents situations where individuals dominate within a group or adjust program code as they go without recognising the value of each individual CT skill (Beecher, 2017; Curzon et al., 2014; University of Canterbury, 2022).

Developing CT learning experiences that reflect the everyday lives and interests of students both inside and outside of the classroom has also been found to increase engagement and motivation. Fox-Turnbull (2016) explains, "there is strong evidence that authentic learning in technology needs primarily to be authentic to culture and practice" (p. 25). This aligns with place-based learning where teachers are encouraged to leverage ākonga interests and lived experiences to develop meaningful in-class tasks (Leonard et al., 2022; Long, 2009). Examples of unplugged contexts that could be used with young children include the creation of word- or picture-based algorithms as instructions for brushing teeth, making playdough, baking a cake, carrying out magic tricks or dances, or instructions for sorting toy cars or soft toys (Curzon et al., 2014; Relkin & Strawhacker, 2021). Curzon et al. (2014) also explore the power of storytelling in developing CT where texts provide a context for a problem that students need to solve. They explain that "contextually rich stor[ies] draw out how a wide range of computational thinking skills are used in an integrated way to solve problems" (p. 90).

When considering the everyday experiences of learners, kaiako should be conscious that the worldviews of the students in their classrooms are frequently different from their own. Creating tasks that reflect students' diverse cultural funds of knowledge has the potential to address inequity where the worldviews and cultural narratives of minority groups may differ from those of dominant cultures within schooling systems (Long, 2009; Relkin & Strawhacker, 2021). Doing so allows ākonga to draw on their personal sociocultural environments and funds of cultural knowledge to support conceptual understanding of CT and create meaning for themselves (Fox-Turnbull, 2016; Hunt & Bers, 2021). This culturally responsive focus engages ākonga in socially just learning that values whakapapa and way of life (Ministry of Education, 2022; Stavrou & Miller, 2017). This is also likely to encourage greater representation of Indigenous people and minority groups within CT pathways in education and related industries in the future (Relkin & Strawhacker, 2021). For New Zealand educators, this involves engaging in practices associated with tātaiako competencies such as whanaungatanga and tangata whenuatanga; through nurturing relationships with Māori learners, whānau, hapū, iwi, and the Māori community to seek guidance around contexts for CT learning experiences that reflect and respect local knowledge, tikanga, stories, and worldviews, and ultimately affirm the culture and identify of ākonga Māori (Education Gazette editors, 2020; Fox-Turnbull et al., 2021).

CT learning experiences incorporating stories of Aotearoa

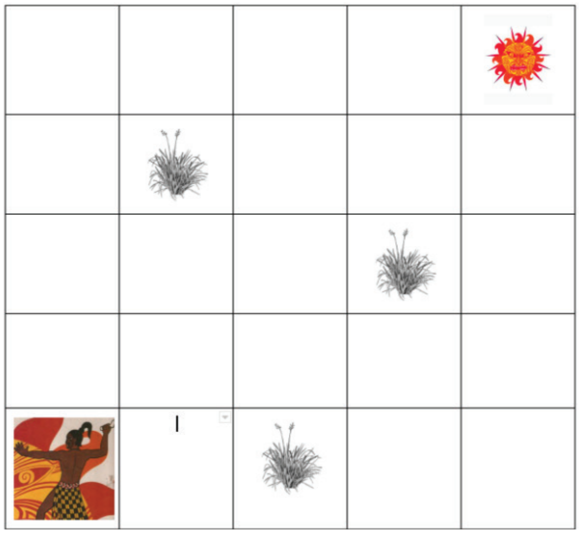
Below are two CTDT learning experiences created by New Zealand educators that were developed with the practices of whanaungatanga and tangata whenuatanga at the forefront.

Learning experience 1: Māui and the sun

The first activity is an unplugged coding task incorporating the pūrākau Māui and the Sun. The activity, inspired by Ministry of Education exemplars (2017), was developed following a speech given to their class by a tohunga whakairo during a marae trip as a collaboration between the second author, Melinda Dixon, and an ākonga Māori. The Māori learner was discouraged to find that many of his classmates were unfamiliar with the stories from te ao Māori that were represented in carvings throughout the wharenui. The teacher asked the ākonga about which stories he liked the most and would like others to understand, suggesting that the stories of Māui be a focus of lessons in the classroom. He was happy with this and provided insight into the pūrākau, positioned as an expert in the classroom. Morals, lessons, and connections to concepts such as seasons found in this pūrākau were importantly explored in kōrero with marae elders while developing this learning experience.

Instructions: Work collaboratively to help Māui collect harakeke to make a rope that will catch and hold Tamanuiterā so that Māui and his brothers can slow the sun and make the days longer. Create a step-by-step set of instructions (algorithm) to move Māui with your group. You will have one programmer who will develop your algorithm, one tester who will move Māui through the board following this algorithm, and one debugger who will identify and fix any errors.

FIGURE 4: MĀUI AND THE SUN UNPLUGGED TASK



Note: From Dixon (2022).

Program example:

→ →
↑ ↑
→
↑
← ←
↑
→ → →

Note: Ask students to put each new instruction on a new line as this aligns with graduating to on-device coding tasks. Orientation of the grid and starting points are also important for this task.

Variations

Whānau and local iwi should be welcomed to share stories of significance to the area as a basis for similar tasks if they deem them appropriate to be utilised in teaching CTDT. Students could also create their own version of this task that aligns with other stories from cultural groups present within the classroom.

Learning experiences 2: A story of Horouta

The second CTDT activity to be shared utilises cross-curriculum connections with the Aotearoa New Zealand's Histories Curriculum to retell the story of Horouta, the first waka to reach Tūranganui-a-Kiwa / Gisborne more than 700 years ago. The activity is based on an assessment by a Māori student teacher R. Hoare in the Graduate Diploma of Teaching (Primary) at Massey University, with permission to share in *Set*. When discussing the creation of this learning experience, Hoare (2022) explains, "a big thing for me is knowing that future ākonga will grow up knowing the stories of their rohe and ancestors" (p. 2). Hoare developed this learning experience for students in her local community in Gisborne, many of whom she was aware could trace their ancestry back to the waka Horouta. Hoare found that there were multiple variations of this pūrākau in existence; therefore, she worked in partnership with the local rūnanga iwi to share the version of the pūrākau (below) that aligns with tūpuna knowledge within the area.

Instructions: Work in a group to create a program that replicates the journey of waka Horouta from Hawaiki to Tūranganui-a-Kiwa. As this was a long journey, you will need to break the voyage into two to three smaller chunks (decomposition). Programmers will be responsible for developing sub-algorithms (algorithmic thinking) to retell their part of the journey. Sub-algorithms will then be combined, and your Tester will run your program by moving your waka through your algorithm while your Debugger identifies and fixes any errors. Finally, groups will share their algorithms and modify these as needed for efficiency (iteration).

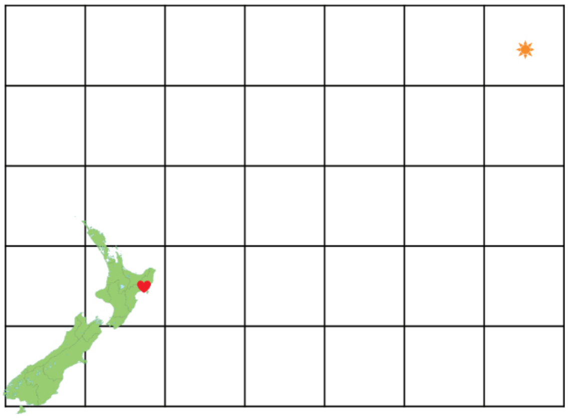
FIGURE 5. A STORY OF HOROUTA

A Story of Horouta

Many people say *Horouta* was the first waka to land in our region more than 700 years ago. On board were Pāoa, who was the captain, Kiwa, who was a tohunga and the navigator of the waka, and Hinehakirirangi, the kaitiaki of the kumara, who was searching for a place to grow her crop. It also had a large crew of men, women and children. After a long journey from Hawaiki to Aotearoa, *Horouta* travelled to Ohiwa, in the Bay of Plenty. Here it hit a sandbar, and the haumi of the waka (the extension and headpiece) was smashed. Pāoa and a group of the crew decided to go to land and search for timbers to fix the waka while Kiwa made minor repairs and continued to navigate the waka to Tūranganui-a-Kiwa (Gisborne) with only a few of the crew so that it could float. Kiwa waited for the land voyagers to arrive on the west bank of the Tūranganui River, at a place he named Tūranganui-a-Kiwa ('the long waiting place of Kiwa'). When Hinehakirirangi arrived, she found fertile soils, planted the kumara, and grew them for future generations.

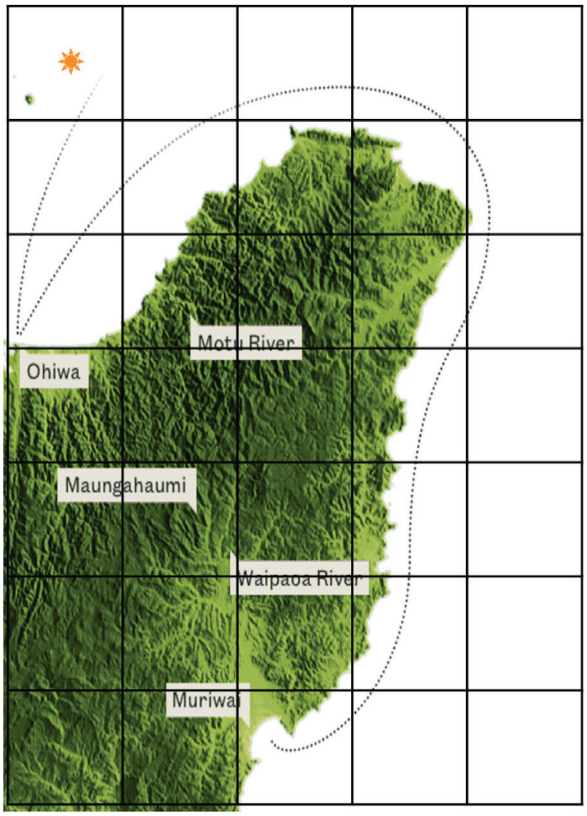
Note: From Hoare (2022).

FIGURE 6. HOROUTA TASK 1 MAP OF AOTEAROA



Note: From Hoare (2022).

FIGURE 7. HOROUTA TASK 2 MAP OF TE TAI RĀWHITI



Note: From Hoare (2022).

Conclusion

This article has explored various approaches for teaching CT at lower levels of the curriculum and the impact of introducing unplugged activities before introducing computerised tasks. This builds on several studies where teachers found that unplugging CT tasks were simpler to implement than first imagined, could be entwined with other curriculum areas, and helped to develop teacher confidence in teaching CTDT. In addition, suggested teaching strategies and examples of learning experiences that incorporate these were shared to support the implementation of unplugged learning experiences in CTDT. These approaches include developing tasks that require explicit use of individual CT skills to solve, assigning student roles that reflect those of the DT industry, and working in partnership with mana whenua and the local community around contexts that reflect and respect the diverse world views of learners.

Mā te kimi ka kite, mā te kite ka mōhio,
mā te mōhio ka mārama.

Glossary

harakeke flax
Tamanuiterā the sun
tohunga whakairo master carver

Notes

1. Other authors, such as Huang and Looi (2021) and Li et al. (2020) support this finding, suggesting that there is an extensive belief that CTDT can be solely linked to activities undertaken on devices.

References

Akiba, D. (2022). Computational thinking and coding for young children: A hybrid approach to link unplugged and plugged activities. *Education Sciences*, *12*(11), 793 https://doi.org/10.3390/educsci12110793

Astrachan, O. (2009). *A new way of thinking about computational thinking*. ACM SIGCSE.

Beecher, K. (2017). *Computational thinking: A beginner's guide to problem-solving and programming*. BCS, The Chartered Institute for IT.

Bell, T., & Vahrenhold, J. (2018). CS unplugged—how is it used, and does it work? In *Adventures between lower bounds and higher altitudes* (pp. 497–521). Springer. https://doi.org/10.1007/978-3-319-98355-4_29

Butler, D., & Leahy, M. (2021). Developing preservice teachers' understanding of computational thinking: A constructionist approach. *British Journal of Educational Technology*, *52*(3), 1060–1077. https://doi.org/10.1111/bjet.13090

Calder, N., & Rhodes, K. (2021). *Coding and learning mathematics: How did collaboration help the thinking?* Mathematics Education Research Group of Australasia. https://www.merga.net.au/common/Uploaded%20files/Annual%20Conference%20Proceedings/2021%20Annual%20Conference%20Proceedings/4_RP_3_Calder_complete.pdf

Crow, T., Luxton-Reilly, A., Wünsche B. C., & Denny, P. (2019). Resources and support for the implementation of digital technologies in New Zealand schools. *ACM International Conference Proceeding Series.* https://doi.org/10.1145/3286960.3286969

Curzon, P., Peckham, J., Taylor, H., Settle, A., & Roberts, E. (2009, July). Computational thinking (CT) on weaving it in. In *Proceedings of the 14th annual ACM SIGCSE conference on innovation and technology in computer science education* (pp. 201–202). https://doi.org/10.1145/1562877.1562941

Curzon, P., McOwan, P. W., Plant, N., & Meagher, L. R. (2014, November). Introducing teachers to computational thinking using unplugged storytelling. In *Proceedings of the 9th workshop in primary and secondary computing education (WiPSCE '14)* (pp. 89–92). https://doi.org/10.1145/2670757.2670767

Digital Skills Forum. (2021). *Digital skills Aotearoa: Digital skills for our digital future.* https://nztech.org.nz/wp-content/uploads/sites/8/2021/01/Digital-Skills-Aotearoa-Report-2021_online.pdf

Dixon, M. (2022). *278413 curriculum and pedagogy. [Course material.]* Institute of Education, Massey University.

Duncan, C. (2018). Reported development of computational thinking, through computer science and programming, and its benefits for primary school students. In *Proceedings of the 49th ACM technical symposium on computer science education* (pp. 275). https://doi.org/10.1145/3159450.3162325

Education Gazette editors. (2020). Combining Māori and digital worlds. *Education Gazette, 99*(20). https://gazette.education.govt.nz/articles/combining-maori-and-digital-worlds/#jumpto-stories-of-teachers-connecting-learners-to-authentic-contexts

Education Review Office. (2019). *It's early days for the new digital technologies curriculum content*. Author. https://ero.govt.nz/our-research/its-early-days-for-the-new-digital-technologies-curriculum-content

Ellington, R., & Leonard, J. (2021). Professional development that fosters computational thinking and high-quality teaching for students of color. In *Fostering computational thinking among underrepresented students in STEM* (pp. 110–137). Routledge. https://doi.org/10.4324/9781003024552-6

Fletcher, G. H. L., & Lu, J. J. (2009). Education: Human computing skills: Rethinking the K-12 experience. *Communications of the ACM*, *52*(2), 23–25. https://doi.org/10.1145/1461928.1461938

Fox-Turnbull, W. H. (2016). The nature of primary students' conversation in technology education. *International Journal of Technology and Design Education*, *26*(1), 21–41. https://doi.org/10.1007/s10798-015-9303-6

Fox-Turnbull, W., Reinsfield, E., & Forret, A. M. (2021). *Technology education in New Zealand: A guide for teachers*. Routledge. https://doi.org/10.4324/9780367816803

Hermans, F., & Aivaloglou, E. (2017, November). To scratch or not to scratch? A controlled experiment comparing plugged first and unplugged first programming lessons. In *Proceedings of the 12th workshop on primary and secondary computing education* (pp. 49–56). https://doi.org/10.1145/3137065.3137072

Hoare, R. (2022). *Digital technology and the New Zealand histories curriculum.* Unpublished assessment. Massey University.

Huang, W., & Looi, C.-K. (2021). A critical review of literature on "unplugged" pedagogies in K-12 computer science and computational thinking education. *Computer Science Education*, *31*(1), 83–111. https://doi.org/10.1080/08993408.2020.1789411

Hunt, L., & Bers, M. U. (2021). Coding, computational thinking, and cultural contexts. In *Teaching computational thinking and coding to young children* (pp. 201–215). IGI Global.

International Society for Technology in Education. (2022). *ISTE computational thinking competencies*. https://www.iste.org/standards/computational-thinking

Leonard, J., Ellington, R. M., Fashola, O. S., Thomas, J., & Mitchell, M. B. (2022). *Fostering computational thinking among underrepresented students in STEM : Strategies for supporting racially equitable computing*. Routledge. https://doi.org/10.4324/9781003024552

Li, Y., Schoenfeld, A. H., diSessa, A. A. et al. (2020). Computational thinking is more about thinking than computing. *Journal for STEM Educational Research*, *3*, 1–18. https://doi.org/10.1007/s41979-020-00030-2

Long, V. M. (2009, September/October). Adding "place" value to your mathematics instruction. *Connect Magazine*, 10–12.

McLennan, D. P. (2017). Creating coding stories and games. *Teaching Young Children*, *10*(3), 18–21.

Metin, S. (2022). Activity-based unplugged coding during the preschool period. *International Journal of Technology and Design Education*, *32*(1), 149–165. https://doi.org/10.1007/s10798-020-09616-8

Ministry of Education. (2017). *Computational thinking for digital technologies: Progress outcomes, exemplars and snapshots*. https://technology.tki.org.nz/Technology-in-the-NZC/CTDT-Progress-outcomes-exemplars-and-snapshots/(tab)/PO2

Ministry of Education. (2022a). *More information on the curriculum change.* https://www.education.govt.nz/our-work/changes-in-education/digital-technologies-and-hangarau-matihiko-learning/more-information-on-the-curriculum-change/

Relkin, E., & Strawhacker, A. (2021). Unplugged learning: Recognizing computational thinking in everyday life. In *Teaching computational thinking and coding to young children* (pp. 41–62). IGI Global. https://doi.org/10.4018/978-1-7998-7308-2.ch003

Rhodes, K. (2020). *The digital technology components in the New Zealand curriculum: Teachers' journey of adoption.* Master's thesis, The University of Waikato. https://hdl.handle.net/10289/13676

Rhodes, K., & Calder, N. (2022). Teachers implementing new primary school digital technology areas: What are we teaching now? *Waikato Journal of Education, 27*(3), 159–173. https://doi.org/10.15663/wje.v27i3.960

Stavrou, S. G., & Miller, D. (2017). Miscalculations: Decolonizing and anti-oppressive discourses in indigenous mathematics education. *Canadian Journal of Education/Revue canadienne de l'éducation*, *40*(3), 92–122.

Sun, D., Ouyang, F., Li, Y., & Zhu, C. (2021). Comparing learners' knowledge, behaviors, and attitudes between two instructional modes of computer programming in secondary education. *International Journal of STEM Education*, 8, 54. https://doi.org/10.1186/s40594-021-00311-1

Te Kete Ipurangi. (2018). *Technology.* https://nzcurriculum.tki.org.nz/The-New-Zealand- Curriculum/Technology/Learning-area-structure#collapsible2

Te Kete Ipurangi. (n.d.-a). *Development paper: Revising the technology learning area to strengthen digital technologies in the New Zealand curriculum.* https://technology.tki.org.nz/content/download/37603/189538/file/DT%20development%20paper%20(final).pdf

Te Kete Ipurangi. (n.d.-b). *Computational thinking for digital technologies: Progress outcomes, exemplars, and snapshots.* https://technology.tki.org.nz/Technology-in-the-NZC/CTDT-Progress-outcomes-exemplars-and-snapshots#:~:text=The%20progress%20outcomes%20describe%20the,computational%20thinking%20for%20digital%20technologies

University of Canterbury. (2022). *CSUnplugged.* https://www.csunplugged.org/en/topics/kidbots/whats-it-all-about/

Withersey, M. (2015). *Start unplugged! Coding in early years.* https://enabling-environments.co.uk/2015/02/02/start-unplugged-coding-in-early-years/

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, *49*(3), 33–35. https://doi.org/10.1145/1118178.1118215

Wing, J. M. (2017). Computational thinking's influence on research and education for all. *Italian Journal of Educational Technology*, *25*(2), 7–14. https://doi.org/10.17471/2499-4324/922

Wohl, B., Porter, B., & Clinch, S. (2015, November). Teaching computer science to 5–7 year-olds: An initial study with Scratch, Cubelets and unplugged computing. In *Proceedings of the workshop in primary and secondary computing education* (pp. 55–60). https://doi.org/10.1145/2818314.2818340

Kate Rhodes is a teaching fellow at the Faculty of Education, University of Waikato.

Email: k.Rhodes@waikato.ac.nz

Melinda Dixon is a professional clinician in the College of Humanities and Social Sciences, Massey University.

Email: M.Dixon1@Massey.ac.nz