International Journal of Technology and Design Education (2026) 36:2101–2119
https://doi.org/10.1007/s10798-026-10085-8

Pedagogical strategies for teaching computational thinking in authentic contexts

Wendy Fox-Turnbull[1] · **Shaoqun Wu**[2]

Received: 13 June 2025 / Accepted: 18 March 2026 / Published online: 30 March 2026
© The Author(s) 2026

Abstract
Computational thinking is important not only in mathematics and computer science but in the other disciplines of STEM and integrated STEM education. This article presents the findings from a study undertaken in a small-town primary school in New Zealand with a high population of Māori students. The study drew on four aspects from Kotsopoulos et al.'s framework of pedagogical experiences: unplugged, tinkering, making, remixing to investigate pedagogical strategies that facilitate the successful embedding of computational thinking within authentic technological practice. Computational thinking is an aspect of digital technologies in the New Zealand Curriculum that teaches children to approach problems systematically, using logical and analytical reasoning. It aims to prepare learners to be designers and developers of digital outcomes. The research identifies learning pedagogies that assist young learners to develop their understanding of computational thinking for the purpose of understanding, designing and developing digital technologies. The qualitative methodologies were informed by the three pedagogical principles from Bishop and Berryman to guide the research design and frame learning support for students. Key themes to emerge from the data include the value of local context, student and classroom organisation, use of physical and digital manipulatives and gamification. Finally, the paper presents a model for the pedagogical delivery of computational thinking when embedded in technological practice in primary schools, drawing from and adding to Kotsopoulos et.al's pedagogical framework.

Keywords Computational thinking · Pedagogy · Teaching strategies · Digital technologies · Technology education · Primary · Authentic technology practice

✉ Wendy Fox-Turnbull
wendy.fox@waikato.ac.nz

[2] School of Computer Science, University of Waikato, Hamilton, New Zealand

[1] School of Education, University of Waikato, Hamilton, New Zealand

Introduction

Computational thinking, an aspect of digital technologies in the New Zealand Curriculum (2017), teaches children to approach problems systematically using logical and analytical reasoning, while also preparing them to become designers and developers of digital outcomes; its importance extends beyond mathematics and computer science to encompass all STEM disciplines and integrated STEM education (Li et al., 2020; Yadav et al., 2016). Computational thinking (CT) is usually understood as a series of skills such as algorithmic thinking, abstraction, decomposition, pattern recognition, generalising, debugging and evaluating ideas and information. These skills are essential not only in the field of technology but also in solving problems in everyday life.

In New Zealand, technology education is based on the principle that students should learn within contexts that are meaningful and relevant to them (Fox-Turnbull, 2003). Unfortunately, many current resources and pedagogical practices focus on the acquisition of decontextualised skills and knowledge with limited links to authentic contexts where students might come to appreciate the relevance of their learning. To address this, it is essential to first understand the cultural context, beliefs, and practices of the community in which teaching and learning occur. Current practices reflect this by incorporating traditional stories and customs into lessons, and by using language that is accessible and culturally appropriate for each specific community. However, we need to go beyond these methods by helping them understand how computational thinking connects to their everyday lives and how it can be applied to solve local issues and challenges. One way to achieve this is by fostering a community-based approach—bringing together local community members, teachers, and students to co-create activities that reflect their needs and interests. Involving community members in the teaching process also helps build trust and a sense of ownership in the learning experience (Berryman et al., 2016). Additionally, it is important to reconsider commonly used tools such as Scratch, which were developed with a bias toward Western modes of thinking and programming. Instead, we should explore tools that support a bottom-up, 'tinkering' approach rather than a linear, top-down style (Bell & Reinsfield, 2012). Hence, there is a need for research that provides examples of how young children can develop computational thinking skills in culturally appropriate and authentic contexts.

This research aims to assist young learners develop foundational computational thinking skills by linking to and drawing on aspects of their cultural knowledge (funds of knowledge) (González et al., 2005) and daily experiences. The study is positioned to connect learning to local environments, communities, and cultures, and encourages schools to tailor their learning programmes by drawing from their unique, local contexts. During this study, two primary teachers and their students co-constructed learning content with the aim of developing computational thinking within authentic contexts where the students could see their cultural understanding and facilitated the creation of a learning space within which computational think is a part of students' everyday lives and culture (Berryman & Forde, 2017; Warren, 2018). To do this a range of pedagogical strategies were deployed to maximise students' learning. This article explores these strategies.

Literature review

Digital technologies in curriculum

The progress of society in various fields (economic, educational, industrial, social) today is strongly coupled to the integration of Information and Communications Technologies (ICT) (Caballero-Gonzalez et al., 2019). The 21st century has transformed the types of skills, knowledge, and competencies needed for success in modern society. Computer science concepts play a significant role across various subjects, and computational thinking is increasingly recognized as an essential skill for everyone. This has led to growing interest in developing computational thinking (CT) from as early as the comprehensive school level. This significant shift toward developing a technology-rich teaching and learning environment has powerful implications for 21st-century education. To broaden student participation in STEM and related careers, robotics and game design have been identified as key focus areas.

The Technology learning area in the New Zealand Curriculum (Ministry of Education, 2007) was updated in 2017 to strengthen its focus on the digital space (Fox-Turnbull, 2019). This update included the introduction of two new technological areas: Designing and Developing Digital Outcomes (DDDO) and Computational Thinking for Digital Technologies (CT) (Ministry of Education, 2017). This development aligned with initiatives in several other countries, including Australia, England, Estonia, and Cyprus (Bell & Duncan, 2015; Duncan & Bell, 2015), and responded to a broad consensus that digital technology is an essential component of 21st-century school curricula (Bell et al., 2014; Falkner et al., 2014; Gander et al., 2013; The Royal Society, 2012). The recommended age to begin learning computational thinking is during primary school, when children are between 5 and 10 years of age (Armoni, 2016; Falkner et al., 2014; Gander et al., 2013; Webb et al., 2017).

Learning aspects of computational thinking and computer programming, rather like that of learning languages are best begun at an early age, at least before the age of 12, but more ideally around eight years of age (Bell & Duncan, 2015; Duncan et al., 2014; Gander et al., 2013; Webb et al., 2017). Students, before the age of 12, form views of their competence in the subject. Students have a natural ability to learning foreign languages before puberty, thus enabling students to be empowered to change the world rather than be passive participants, and it prepares students for a future in computing. In addition, successfully developing computational thinking at lower levels can assist the learning of computer science in the later schooling years and beyond (Fletcher & Lu, 2009; Jacob & Warschauer, 2018). In their meta-analysis, Montuori et al. (2024) identified learning computational thinking at pre and primary school significantly improved problem-solving.

Computational thinking

Computational thinking is an aspect of digital technologies in the New Zealand Curriculum (Ministry of Education, 2017) that teaches children to approach problems systematically, using logical and analytical reasoning. “It is important not only in mathematics and computer science but also across the other STEM disciplines, and it can be fully integrated into STEM education.” (Li et al., 2020; Yadav et al., 2016). Computational thinking is usually

understood as a series of skills that includes algorithmic thinking, abstraction, decomposition, sequencing and generalising patterns, evaluation, and logic applied in the realm of digital technology and everyday life. Introducing computational thinking to young children is an excellent way to set them up for future success in a rapidly changing world (Yadav et al., 2016), however, anecdotal evidence suggests that when young students (aged 5–8) learn such skills they need to do so in meaningful contexts where they are invested in the outcomes and can utilise their prior knowledge and cultural experiences. Unfortunately, many current resources and pedagogical practices focus on the acquisition of decontextualised skills and knowledge with limited links to authentic contexts where students might come to appreciate the relevance of their learning (Fox-Turnbull, 2019).

Pedagogical strategies for teaching digital technologies and computational thinking

There is increasing interest in the teaching of computational thinking skills andknowledge in the compulsory schooling sector. In the primary sector, this is particularly challenging because few teachers are experienced with sound content knowledge (CK) and pedagogical content knowledge (PCK) in CT (Rehmat et al., 2020; Yeni et al., 2021). Given recent developments in digital technologies curriculum, it is increasingly important for teachers to equip themselves with the necessary content and pedagogical knowledge and skills so they can teach their students effectively and ensure they remain engaged (Rehmat et al., 2020). Kandemir et al. (2021) state that quality instruction with appropriate pedagogical approaches is more effective at increasing student achievement in computational thinking than tools, programmes and environments.

Many tools and materials have been developed in the last 10 years to teach children CT, including such things as block-based programming, robotics kits, simulation tools, game-based learning and unplugged activities (Rao & Bhagat, 2024). However, to use these materials and tools effectively, teachers need to understand the key concepts that underpin CT and how to use available resources to best benefit their learners.

Several researchers have identified key concepts and strategies to assist teachers' CT pedagogical knowledge (Rao & Bhagat, 2024). Kong et al. (2020) identify three stages in the learning process for computer programming: understanding the problem; devising a solution; and transforming the solution into a programming language for execution (to play to think, to code). All steps involve computational thinking. Rao and Bhagat (2024) state that CT includes four key ideas: decomposition, breaking down into manageable parts; pattern recognition, finding and naming patterns and trends; abstraction, showing only the important information needed; and algorithmic design, step-by-step instructions. In their exploration of primary school children working in didactic relationships while collaborating on programming tasks Grover et al. (2016)found that children attempted to understand, represent, and then execute their ideas, thus offering some hierarchy to children's learning. Kotsopoulos et al. (2017) offer a framework of pedagogical experiences which identify four aspects with a hierarchy implied shown in Fig. 1: unplugged, tinkering, making, remixing.

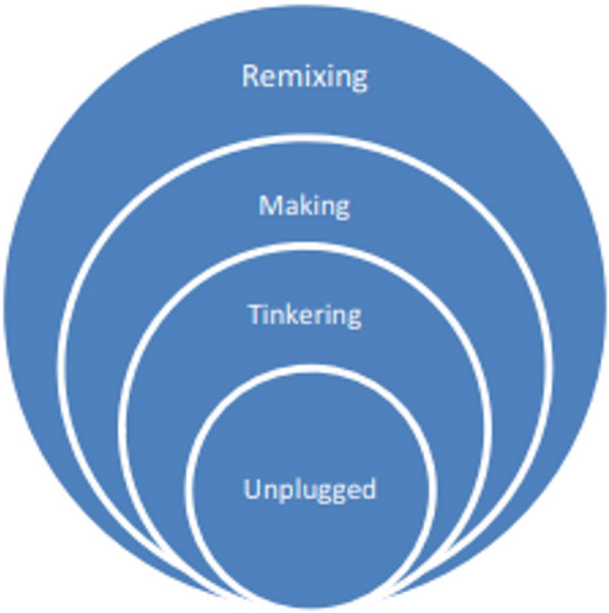


Fig. 1 Four Pedagogical Experiences (Kotsopoulos et al., 2017, p. 159)

Unplugged activities

There are clear benefits for young people when they begin learning computational thinking (CT) in an 'unplugged' environment—that is, without the use of digital devices. Unplugged CT learning eliminates the need for computers and supports a wide range of neurodiverse learners by relying less on text and offering more accessible, hands-on approaches, making it especially suitable for young children. They are easily designed to be collaborative and are excellent for foundational learning (Hermans & Aivaloglou, 2017; Kotsopoulos et al., 2017). Unplugged activities have been found to effectively develop key computational thinking (CT) concepts—sometimes more so than plugged approaches—and can be made academically rigorous (Hermans & Aivaloglou, 2017; Kotsopoulos et al., 2017; Metin, 2022; Wang et al., 2025; Wohl et al., 2015). This is particularly true at the primary level in a game-based environment (Wang et al., 2025). They are engaging, relevant to young learners, and support the development of self-efficacy (Fletcher & Lu, 2009).

Tinkering

The act of tinkering is a critical component in digital technology education. It involves fiddling and playing with technologies, taking them apart, making and exploring changes and or making modifications to components of a whole or the whole itself. "The goal of tinkering experiences is to provide a context for exploring incremental modifications, without the additional cognitively demanding challenge of actually building the object. Application, simulation, and problem-solving are the foci. These experiences ultimately explores the 'what if?'" (Kotsopoulos et al., 2017, p. 161). In the digital space this could involve playing and fiddling with computer programmes, manipulating code building blocks, undertaking

electronic simulations. For example, students can tinker with block code in programs like Scratch to explore how changes affect the avatar's actions. (Kotsopoulos et al., 2017).

Making

Experiential learning theory emphasises the critical role that experience plays in the learning process.". During *making* learners must draw from a full range of learning abilities (Kolb & Kolb, 2009), including problem-solving, identifying and selecting most suitable tools and materials, sequencing necessary tasks, and undertaking construction, while continually evaluating design decisions and outcomes. In the digital space *making* is writing the code to meet a specific need or solve a preidentified issue or problem (Kotsopoulos et al., 2017). Wilkerson-Jerde (2014) states "Constructing computational artifacts encourages students to combine multiple ideas into a cohesive process, organize their understanding in new ways, and 'debug' understandings in their instructions to produce something unexpected" (p. 102). They also suggest publishing students' creations to foster a sense of ownership and to encourage giving and receiving feedback.

Remixing

Remixing involves using combining of already developed outcomes or parts of outcomes in new and creative ways or repurposing them for new and innovative applications (Kotsopoulos et al., 2017). Modification and Adaptation are at the heart of these processes. Because "the student must be able see useable objects within other objects" (p. 165) during *remixing*, Kotsopoulos et al. (2017) suggest that it is the most cognitively demanding experiences, requiring the learner to think at the outer edges or advancement of their zone of proximal development.

Developing strong pedagogical strategies for teaching computational thinking is directly linked to successful learning in the design and development of digital outcomes. Kong et al. (2020) identify a lack of empirical research in the area of developing teacher pedagogical content knowledge and content knowledge in computational thinking. Thus, there is a need for research that provides examples of how teachers support young children in developing computational thinking skills in authentic and culturally appropriate contexts. This article explores pedagogies used to teach computational thinking in two primary classrooms with students aged 8–10 years, in a low-decile semi-urban school in New Zealand, guided by Kotsopoulos et al. (2017) framework of pedagogical experiences.

Research question

What learning pedagogies assist young learners to develop their understanding of computational thinking for the purpose of understanding, designing and developing digital technologies?

Methodology and Methods

This qualitative study was guided by the three pedagogical principles from Bishop and Berryman (2006), which informed both the research design and the framework for student learning support. The co-constructed learning sequences and processes were characterised by:

- being embedded in students' local community (school and local small town).
- innovation and creativity among students, who will be encouraged to be creative in learning by deploying individual culture experiences.
- interactivity, with students developing new knowledge through dialogue with peers, teachers and community members, the teachers' feedback and guidance, and interaction with the co-authoring digital tools.

Gonzales, Moll and Amanti (2005) suggest that valuing students' cultural funds of knowledge and bringing this understanding and knowledge into the classroom enhances students' achievement. This study focused on teaching computational thinking in a meaningful culturally situated context as intended in The New Zealand Curriculum ((NZC) (Ministry of Education, 2017). Skills-based lessons were delivered on a need-to-know basis and embedded within authentic, culturally relevant contexts connected to students' current lives and potential futures (Hennessy, 1993).

Tools and resources

The ByteEd *Dino Steps* kit[1] was used to introduce foundational computational thinking (CT) concepts such as sequencing, orientation, decomposition, algorithmic thinking, and debugging. Students engaged in activities that required them to break down tasks into step-by-step instructions, learning to decompose problems and create clear sequences of actions. By laying out directional cards—such as "forward," "turn left," and "turn right"—they planned paths for Explorer Ed to follow, practicing orientation and spatial awareness. These sequences formed simple algorithms that guided Ed across grids or maps, allowing students to test and refine their thinking. When Explorer Ed did not reach the intended destination, students applied debugging strategies to identify and correct errors in their instructions.

Kete, a mobile application developed by the research team, was used to reinforce students' understanding of computational thinking (CT) concepts and coding in a meaningful, context-rich environment. Figure 2a show the home screen of *Kete*. Central to 'Kete' is the Poutama motif, an element in Māori culture. Poutama, traditionally a stepped pattern in tukutuku panels and woven mats, symbolises genealogies and levels of learning and intellectual achievement. It echoes the journey of Tāne-o-te-wānanga ascending to the heavens in pursuit of knowledge. Another important feature of the Kete App is that the students are able to choose which language they use within the App, English or Te reo Māori [Indigenous language of New Zealand]. Figure 2b shows the page where students select the language they wish to use.

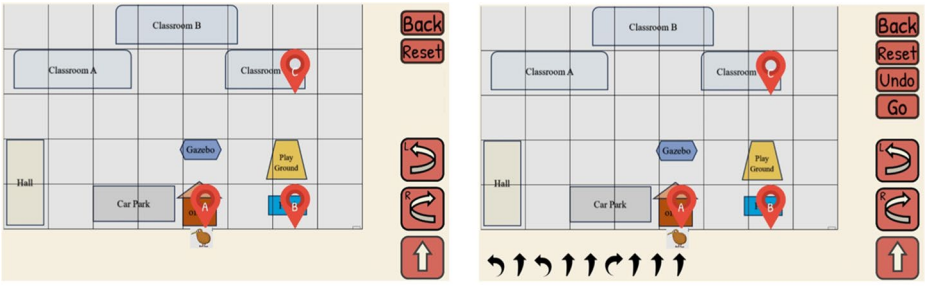
As shown in Fig. 3a, to play the game, the player clicks the directional arrows (left, right, and forward) on the right panel to direct the Kiwi (a native New Zealand bird) icon to navi-

[1] The kits were developed by a New Zealand company, ByteEd and given to the school for free use.



(a) (b)

Fig. 2 Kete app home page. **a** Kete's home screen. **b** Kete's Settings



(a) (b)

Fig. 3 **a** FreePlay and Tour mode. **b** Sequence Mode

gate to the locations marked A, B, and C. Game completion is achieved by successfully visiting all locations. *Kete* offers three difficulty levels through three playing modes. FreePlay Mode allows players to visit the locations (A, B, C) in any order. Tour Mode requires the player to visit the locations in a specific sequence—A-B-C—thus promoting logical thinking and planning skills. Sequence Mode, as shown in Fig. 3b, requires the player to program a sequence of instructions (arrows) by clicking the arrow icons on the right panel or using the "undo" button to delete the previous instruction. Clicking the "go" button executes the program, which is displayed below the map, directing the Kiwi to visit the locations in the A-B-C sequence.

Kete allowed students to design and create their own games through the game creation interface shown in Fig. 4 where an image, the Google satellite view of the school, is uploaded as the background map and a grid is overlaid. The size of the grid (the number of rows and columns) can be adjusted using the sliding bar on the right. Each grid square in this grid represents a position of a location. A location, along with its text, audio description, or a detailed images about the locations then can be added to the map as shown in Fig. 4a. The text, audio and images are in a pop-up window when the player navigates to the location. Figure 4b shows five locations A-E are added to the game map.

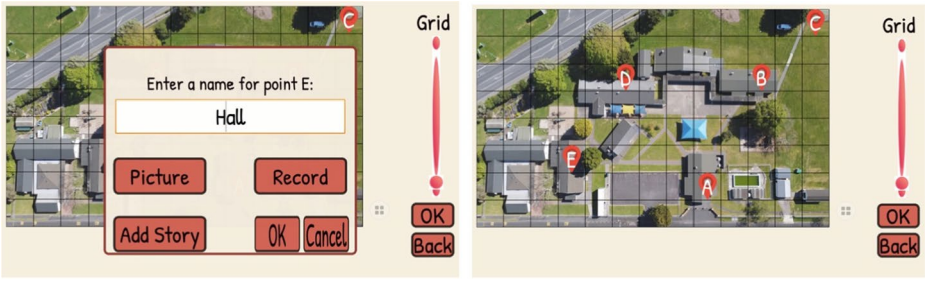


Fig. 4 The game creation interface (**a**) adding picture, text and audio to a location. **b** five locations A-E are added to the game map

Participants and ethical approval

The team investigated pedagogies that led to successful student learning in digital technology. It drew on findings from previous study which investigated sequencing and orientation aspects of computational thinking using Beebot® robots, with 35 Year 0–3 students, which highlighted the success of using te reo Māori, a Tuakana/Teina teaching approach and merging lived experiences as Māori with technology education (Fox-Turnbull et al., 2024).

In the study reported in this article ethical approval was gained through the University of Waikato Division of Education ethics committee. Participation was voluntary and consent was gained by the school principal, participant teachers, students and their parents. Confidentiality was guaranteed however anonymity was not due to the limited number of schools which fit the description within the region. Methods used in the study included initial and final semi-structured interviews with teachers, initial interviews with 10 consented students and researcher observations of teaching and learning with related photographs.

Procedure and classroom

This study took place in two primary classrooms in a small-town low decile school in New Zealand (NZ) over four weeks. The first was Year 4 class with a first-year teacher, Whaea K, the second a Year 5–6 class taught by Whaea O. Teachers' names are pseudonyms. 'Whaea' is a Māori term for respected woman and often used in NZ primary schools. The research was undertaken by two researchers with assistance from two student summer scholars. The first researcher was a former primary teacher and current academic in technology education, the second a computer science lecturer. The researchers observed that Māori culture was heavily reflected in their classroom. Most of the students in the classroom and all the student participants self-identifed as Māori. Te Reo Māori was evident on the classroom walls. The students were guided by Māori values, seen on the wall in their classroom in Fig. 5.



Fig. 5 Whaea O's classroom

The context used in this study included the students' school and their hometown. When designing a school tour with *Kete*, students were sent off in groups around their school to determine the important information and locations that would be needed for the app. Figure 6 show examples of the students' school in the Kete app, which so engaged the students.

During the first three weeks (one hour per day), students were introduced the basic computational concepts of sequencing, algorithm, abstraction, pattern recognition and debugging with Dinosaur Steps kits. The kits are simple and give the students a way to code a sequence of steps on a grid map without the need of writing code down even if their literacy and numeracy skills are not advanced and fully developed. In the last week, teachers divided students from different years with different literacy and numeracy skills into groups, eight to ten students in each group. The research team supervised the students in each group who used Kete to design and develop a digital tour of their school for newcomers.

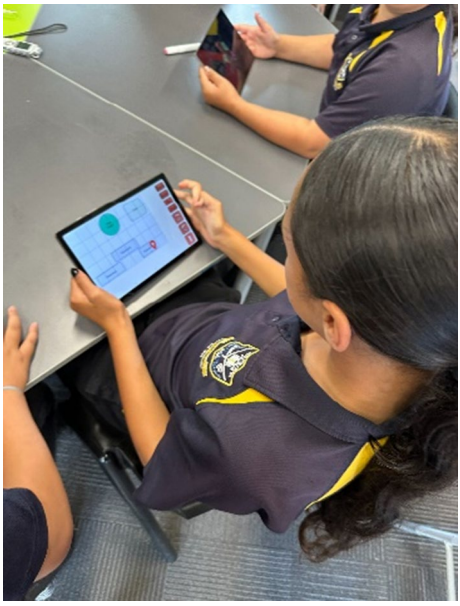


Fig. 6 Students with Kete app map of the school

A framework was developed from the interview schedule and the research objectives and used for initial coding and sorting of the data. Thematic analysis was applied to elicit key themes. Braun and Clarke (2006, p. 78) consider this a "foundational method of qualitative analysis … independent of theory and epistemology and can be applied across a range of theoretical and epistemological approaches". Representative quotes were identified and are presented in the findings. Rigour and consistency were ensured by one summer scholar completing initial coding, which was then cross-checked by the two researchers independently to identify the initial themes. This article reports on key aspects of the successful pedagogical strategies theme, which was further divided into subthemes for future analysis.

Findings

Key themes that emerged from the data related to successful pedagogical studies included the (1) value of local context while undertaking authentic technology practice, (2) student and classroom organisation, (3) use of physical and digital manipulatives.

Value of local context embedded in technology practice

A key observation made by teachers and researchers was that student engagement appeared to flourish when students were able to apply their lived experience to lessons. When asked where she observed learner success Whaea O said "Applying their own context to their [programmes] and being able to create their own mix". Being Māori and local sites around their village were some of the more noticeable engagement areas and exemplified when the students were introduced to pattern recognition. When asked about the patterns they see one child stated they saw patterns "in the marae [tribal communal area]". See Fig 7. for an example of some of the patterns commonly seen in marae.

The following extract from the researcher Observation explains this in more detail.

> The class were brought back together, and everyone had an opportunity to share their ideas. WO [teacher participant] focused on the student who said the marae and asked the students where they could find these patterns. The students said in the carvings and WO pointed out additional patterns in the tukutuku panels in the wharenui [meeting house on the marae] and a key feature they possessed was that the patterns repeated themselves (Researcher Observation p10).

When teaching the concept of abstraction, the students were shown to maps from their local area. The first a google map of their village and the second one of their schools showing the different facilities. Researcher observations indicated student motivation and excitement. "Student were excited to identify different places they recognised such as the shops they frequented and streets they lived on" (Researcher Observation p7). Whaea K stated, "It was cool to have [them] create their own maps, and I definitely saw them thinking about their own community, putting in their own shops, putting in the fruit shack".

This finding was further illustrated in reverse when the students were confronted with a cross section diagram of a house that had features they were not familiar with such as an attic and garage and were asked to locate technologies within in.



Fig. 7 Tukutuku (inner) and Kowhaiwhai (outer) Patterns

> The house had an attic which confused some of the students so most guessed different things to find in there. Some students struggled with the concept of a garage as well, mostly confusing it for a carport and did not think technology belonged outside (Researcher Observation 6 Nov).

Another indicator of student engagement was evident during the post-unit focus group interview. When asked about how they see computing in their future, it became clear that students had learned they could become creators of digital technologies, not just users. "Make my own app and work on it properly and see if it's the right app to put in on the public" and "I want to make an app for little kids who don't know anything, and they can write a little code and learning everything" are two quotes that exemplify this.

Classroom and student organisation

An important aspect of this research project was for teachers to employ a diverse range of organisational practices to meet the needs of their students. This was achieved by altering between whole class, group, and individual learning experiences. In the example below learning was introduced with a whole class session led by Whaea O. Within the whole class discussion Whaea O directed her students to discuss ideas with those around them before sharing their ideas with the whole class.

> We're thinking about our digital devices and the world, I want you now to turn to the person near you or next to you and think about what devices do you have at home and how do they help you. What are some digital devices do you have at your house and how do they help you turn and talk to your friends?

During the unit the students also worked in small, teacher-selected groups, for example when asked to identify about technology they might find at home.

> Students were then split into groups and given a task to complete. The students were presented with a blank cross section of a house and asked to identify different types of technology found in the home and where they might be located (Researcher Observation 6 Nov).

The unit continued with a range on unplugged activities through in Dino Steps. When learning through the Dino Step the students worked in teacher-selected groups who need to put together a prehistoric landscape jigsaw, then programming Explorer Ed about a prehistoric landscape and through a range of challenges related to the dinosaur inhabitants. Figure 8 shows an example of one of the activity sheets provided as a teacher resource with the kits where the students worked within their small groups through a series of challenges aimed at developing basic computational thinking skills.

At times the students worked as individuals. This was the case later in the unit when the student each engaged in Kete to design a 'visitor guide for their school'. The students designed and developed their own icon in Kete and were given autonomy to design their



Fig. 8 A ByteEd Dino Steps Activity Sheet

own pathway around the school. In her exit interview when asked about her highlights of the unit Whaea K said "I like how they could be so independent with the learning". Whaea O elaborated

> And they didn't wait on their mates….It's quite good to give them the independence where they can challenge themselves and go as far as they can go. [Test] their capabilities where they're not held back by any of their peers.

Use of non-digital and digital manipulatives

When first learning computational thinking concepts the students were presented with physical manipulatives to help extend their thinking and cement different concepts. In Dino Steps these included the jigsaw place map, Explorer Ed, dinosaurs, and the arrow cards for creating the code as seen in Fig. 8. The manipulatives proved invaluable for students with lower levels of literacy and numeracy, helping them understand concepts such as sequencing and code notation, and reducing learning barriers.

> The visual, the visual application of it, like the dinosaurs, the creating the map, the challenges. They really liked the challenges and the resources like the step cards and the arrows. It was really good for them, clear, easy to use. And I feel like if we had that with the juniors, I'd pick it up a lot easier. It was easy for them to pick up (Whaea O Exit Interview).

Non-digital manipulatives also allowed the students to be creative and explore concepts at their own pace and ability. With teacher conferencing, students' thinking was extended, thus assisting the development of complexity in coding.

> Students were then sent back to their groups with the Dino Step kits and were instructed to create a sequence of instructions that repeated itself at least twice. They were told that the pattern could be as simple or as complex as they wanted, the only criteria they were given was that it needed to repeat itself twice (Researcher Observation 22 Nov).

Transference from physical to digital manipulatives was assisted through the use of a common concept between Dino Steps and *Kete*. Both used a grid requiring students to write and debug algorithms for journeys through an environment, Dino Steps in the physical space and Kete digital. Researcher observation indicated the switch from physical to abstract coding proved challenging at first. "This was one of the first times that students had to code without using manipulatives. Some students struggled to get started but with some teacher support, they began to get the concept" (Researcher Observation 21 Nov).

An interesting finding this that when students were working on Kete debugging proved challenging for some students as when an error occurred as they initially blamed the app for errors, "It didn't do what I told it to do" (frustrated student). However, the data suggests that those who were able to understand that the fault was not the app but their programme, were able to debug their programme and went on to become more proficient at coding. "Students who can debug effectively became more proficient at coding" (Observations, Researcher Journal).

Discussion

Learning concepts of computational thinking was a challenging topic for both teachers and students in this study. During initial interviews teachers expressed feelings of being underprepared to teach as they had a minimal grasp on computational concepts themselves. This finding aligns with those of Rehmat et al. (2020) that primary teachers were challenged by computational thinking content knowledge. As teachers deepened their own understanding and added to their own pedagogical practices, they felt more confident in the content they were teaching. Kandemir et al. (2021) and Yeni et al. (2021) found that increasing teachers' content and pedagogical knowledge led to improved student learning. To assist teachers to develop their pedagogical strategies for teaching computational thinking the Kotsopoulos et al. (2017) model goes some way to assisting teachers. The findings of this study have prompted a number of modifications to the Kotsopoulos Model as shown in Fig. 9.

The most notable modification involves embedding Kotsopoulos and colleagues' model within the context of authentic technology practice. The need for an authentic context for technology education is well documented (Fox-Turnbull, 2003; Fox-Turnbull et al., 2023; Hill, 2017; Snape & Fox-Turnbull, 2013). In this framework, context provides both direction and meaning for all learning activities. Teachers embedded student learning in authentic technological context, with a purpose that was meaningful to the students. The data from this study, for example, the students were excited to see their school and community represented in the Kete App, highlight this as essential for fostering student engagement and

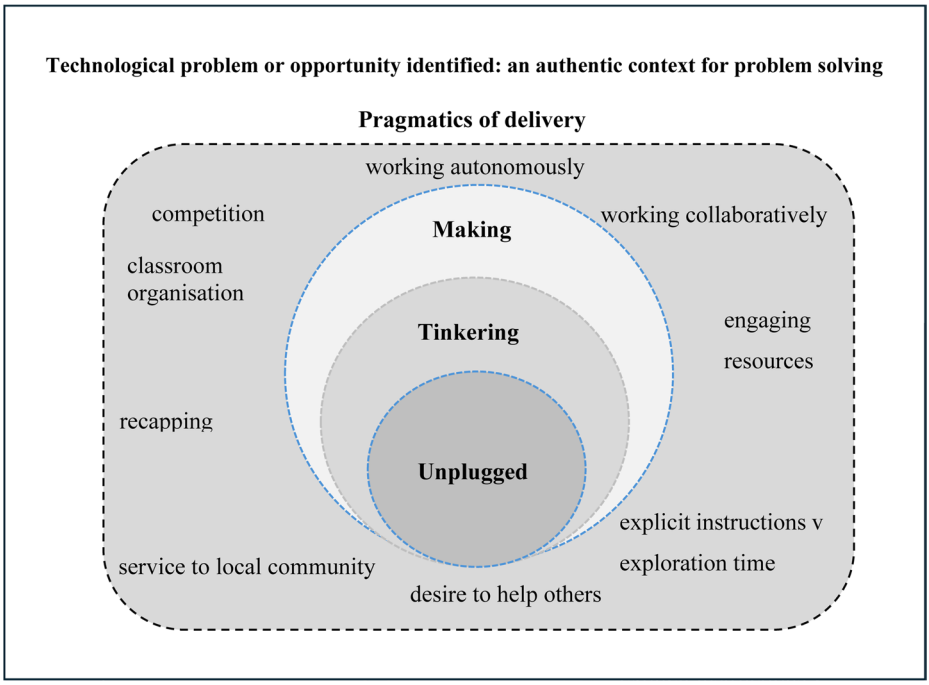


Fig. 9 Pedagogical Delivery of Computational Thinking within Technological Practice in Primary Schools. (modified from Kotsopoulos et al., 2017, p. 159)

motivation. Additionally, the model demonstrates that authentic technological practice in primary classrooms is largely shaped by 'the pragmatics of delivery'. These, listed in Fig. 9, include a range of instructional strategies assist students' development of computational thinking skills and knowledge (Kandemir et al., 2021, and a range of organisational strategies to ensure all students are engaged and learning (Rehmat et al., 2020. In this study, such strategies encompassed revisiting prior learning, offering explicit instructions, allowing student autonomy in exploring materials, incorporating unplugged and online resources, fostering healthy competition, and providing opportunities for students to make meaningful contributions. Class and students organization strategies include students working autonomously, in small groups or as a whole class and teachers took time to recap earlier learning.

Kotsopoulos et al. (2017) suggest several advantages of starting student learning in computational thinking with unplugged activities. Students can develop basic to complex programming concepts and skills through unplugged activities (Bjursten et al., 2023; Léonard et al., 2019). In this study Dino-Steps proved successful in developing the students' understanding of sequencing, orientation, decomposition, abstraction, notating algorithms and debugging. Due to the varying academic abilities of students, students in this study faced practical and conceptual challenges. The conceptual skill that students most struggled understand was abstraction. Teachers noted that this may have had to do with students recognizing different things as important, even though from an adult perspective, they were not. Regarding practical challenges, the manipulatives such as Dino-Steps proved to be invaluable as it possible for students with low literacy skills to learn computational thinking concepts without relying on pen and paper.

Tinkering allowed the students to make mistakes in a low-risk environment. As the students moved to the digital space the students were given the freedom to play (tinker). The tinkering phase allowing this understanding to develop, with the assistance of teachers and peers when needed. When tinkering they were only accountable to themselves and therefore at low risk. In the study, students who began to understand abstraction and debugging progressed to the point that they began to actively debug during their coding. Understanding of the need to debug coding occurred naturally for some students but was challenging for others, especially if they assumed the fault was with the app rather than their code. As their understanding progressed students were challenged with specific coding tasks, therefore moving in to the 'making' phase of their learning (Kong et al., 2020).

Another modification includes a subtle shift from 'Making' to 'Designing and Making'. In this study, students were presented with an 'issue or problem' situated in an authentic context, which they addressed by designing and creating a potential solution. As a result, the process evolved from simply 'making' to 'designing and making'. This change is especially relevant given the New Zealand Curriculum, which places digital technologies within the broader Technology learning area (Fox-Turnbull, 2019). Consequently, students are encouraged to develop computational thinking skills purposefully, aiming to become creators—not just users—of digital technologies (Ministry of Education, 2017 b).

The final key change to Kotsopoulos and colleagues' model is the omission of 'remixing'. This component was excluded in Fig. 9 because the primary-aged students involved in the study had not yet developed the level of sophistication necessary for remixing.

Conclusion

This article presents a modified and expanded model of pedagogical delivery of computational thinking that is embedded within authentic technological practice in primary schools. All too often computational skills and concepts are taught in isolation. This study highlights the significant role of local context, diverse pedagogical strategies, and the integration of both physical manipulatives and digital tools in fostering successful computational thinking skills among primary school students. The findings underscore that when students connect their learning to their own lived experiences—such as their community, cultural background, and familiar local environments—they become more engaged and motivated in their technological learning. By embedding lessons in authentic contexts, students were able to apply computational thinking concepts such as abstraction, decomposition, pattern recognition, sequencing, and debugging in meaningful and relevant ways.

The research also demonstrates the importance of employing a variety of organizational strategies, ranging from whole-class discussions to autonomous group work and individual tasks. These approaches ensured that all students, regardless of their literacy or numeracy levels, were able to engage with and develop an understanding of the concepts involved in computational thinking. Physical manipulatives, such as the Dino-Steps kits, played a crucial role in helping students build foundational knowledge, while the digital tools Kete allowed students to explore and apply their learning in more abstract but real-world contexts. The transition from physical to digital manipulatives, although challenging at first, helped students deepen their understanding of computational concepts.

The study also emphasizes the critical role of teacher development in supporting effective pedagogical delivery of computational thinking. As teachers gained confidence and understanding in the subject, their ability to adapt their teaching strategies to meet students' needs led to enhanced student learning outcomes. Despite initial challenges, the students' ability to debug and problem-solve improved as they advanced in their learning journey, illustrating the transformative power of a hands-on, contextually rich, and student-centered approach to teaching computational thinking.

In conclusion, this research offers valuable insights into the effective integration of technology and computational thinking in primary education. By continuing to innovate pedagogical strategies and leveraging the power of local context and manipulatives, educators can help cultivate critical digital skills that prepare students for future challenges in a technology-driven world.

Funding Open Access funding enabled and organized by CAUL and its Member Institutions.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

Armoni, M. (2016). Computer science, computational thinking, programming, coding: the anomalies of transitivity in K-12 computer science education [Opinion]. *ACM Inroads, 7*(4), 24–27.

Bell, B., & Reinsfield, L. (2012). Becoming a culturally responsive teacher New Zealand Association for Research in Education Conference and Annual Conference, Hamilton, New Zealand.

Bell,T., Andreae, P., & Bobins, A. (2014). A case study of the introduction of computer science in NZ schools. *ACM Transactions on Computing Education, 14*(2), 10.10–10.31. https://doi.org/10.1145/2602485

Bell, T., & Duncan, C. (2015). Proposed digital technologies curricula up to NZC level 5 [Report for the Place of Digital Technologies in NZC reference group].

Berryman, M., Egan, M., & Ford, T. (2016). Examining the potential of critical and Kaupapa Māori approaches to leading education reform in New Zealand's English-medium secondary schools [Journal Article]. *International Journal of Leadership in Education*, 1–14. https://doi.org/10.1080/13603124.2016.1206973

Berryman, M., & Forde, T. (2017). Culturally responsive relationships promoting partnerships between schools and indigenous Māori families and communities. In R. McNae, & B. Cowie (Eds.), Realising innovative partnerships in educational research (pp. 147–156). Sense.

Bishop, R., & Berryman, M. (2006). Culture Speaks: Cultural relationships and classroom learning. Huia.

Bjursten, E. L., Nilsson, T., & Gumaelius, L. (2023). Computer programming in primary schools: Swedish Technology Teachers' pedagogical strategies. *International Journal of Technology and Design Education*, *33*(4), 1345–1368. https://doi.org/10.1007/s10798-022-09786-7

Caballero-Gonzalez, Y. A., Muñoz-Repiso, A. G. V., & García-Holgado, A. (2019). Learning computational thinking and social skills development in young children through problem solving with educational robotics Proceedings of the Seventh International Conference on Technological Ecosystems for Enhancing Multiculturality, León, Spain. https://doi-org.ezproxy.waikato.ac.nz/10.1145/3362789.3362874

Duncan, C., Bell, T., & Tanimoto, S. (2014). November). Should your 8-year-old learn coding? (pp. 5–7). WiPSCE.

Falkner, K., Vivian, R., & Falkner, N. (2014). The Australian digital technologies curriculum: challenge and opportunity. Australasian Computing Education Conference (ACE), Auckland, New Zealand.

Fletcher, G. H. L., & Lu, J. J. (2009). Education: Human Computing Skills: Rethinking the K-12 experience. *Communications of the ACM*, *52*(1), 23–25. https://doi.org/10.1145/1461928.1461938

Fox-Turnbull, W. (2003). The place of authentic technological practice and assessment in technology education [Thesis for Master of Teaching and Learning, Christchurch College of Education]. Christchurch.

Fox-Turnbull, W. H. (2019). Implementing digital technology in the New Zealand Curriculum. *Australasian Journal of Technology Education*.

Fox-Turnbull, W. H., Wu, S., Stafford, M., & Mayo, T. (2023). Embedding Computational Thinking into Authentic Technology Practice. The 40th International Pupils' Attitudes Towards Technology Conference Proceedings 2023.

Fox-Turnbull, W., Wu, S., Mayo, T., & Stafford, M. (2024). Computational thinking for young indigenous learners in New Zealand. *Future in Educational Research*, (2), 131–146. https://doi.org/10.1002/fer3.29

Gander, W., Petit, A., Berry, G., Demo, B., Vahrenhold, J., McGettrick, A., Boyle, R., Drechsler, M., Mendelson, A., Stephenson, C., Ghezzi, C., & Meyer, B. (2013). Informatics education: Europe cannot afford to miss the boat. Report of the joint informatics Europe & ACM Europe working group on informatics education. I. E. A. E. W. Group.

González, N., Moll, L. C., & Amanti, C. (Eds.). (2005). Funds of knowledge (1st ed., Vol. 2009 Reprint). Routledge.

Grover, S., Bienkowski, M., Tamrakar, A., Siddiquie, B., Salter, D., & Divakaran, A. (2016). Multimodal analytics to study collaborative problem solving in pair programming. Proceedings of the sixth international conference on learning analytics & knowledge.

Hennessy, S. (1993). Situated cognition and cognition apprenticeship: implications for classroom learning. *Studies in Science Education*, *22*, 1–41.

Hermans, F., & Aivaloglou, E. (2017). To scratch or not to scratch? A controlled experiment comparing plugged first and unplugged first programming lessons. The 12th workshop on primary and secondary computing education.

Hill, A. M. (2017). Authentic Learning and Technology Education. In de M. Vries (Ed.), *Handbook of Technology Education* (pp. 473–488). Springer.

Jacob, S. R., & Warschauer, M. (2018). Computational thinking and literacy. *Journal of Computer Science Integration*, *1*(1). https://doi.org/10.26716/jcsi.2018.01.1.1

Kandemir, C. M., Kalelioğlu, F., & Gülbahar, Y. (2021). Computational thinking: Enhancing STEAM and engineering education, from theory to practice. *Computer Applications in Engineering Education*, *29*(1), 29–45. https://doi.org/10.1002/cae.22374

Kolb, A. Y., & Kolb, D. A. (2009). Experiential learning theory: A dynamic, holistic approach to management learning, education and development. *The SAGE handbook of management learning education and development*, *7*(2), 42–68.

Kong, S. C., Lai, M., & Sun, D. (2020). Teacher development in computational thinking: Design and learning outcomes of programming concepts, practices and pedagogy. *Computers & Education*, *151*, 103872. https://doi.org/10.1016/j.compedu.2020.103872

Kotsopoulos, D., Floyd, L., Khan, S., Namukasa, I. K., Somanath, S., Weber, J., & Yiu, C. (2017). A Pedagogical Framework for Computational Thinking. *Digital Experiences in Mathematics Education*, *3*(2), 154–171. https://doi.org/10.1007/s40751-017-0031-2

Li, Y., Schoenfeld, A. H., diSessa, A. A., Graesser, A. C., Benson, L. C., English, L. D., & Duschl, R. A. (2020). Computational thinking is more about thinking than computing. *Journal for STEM education research*, *3*, 1–18.

Léonard, M., Peter, Y., & Secq, Y. (2019). Patterns and Loops: Early Computational Thinking. Transforming Learning with Meaningful Technologies: Technology Enhanced Learning, EC-TEL 2019, Delft, The Netherlands.

Metin, S. (2022). Activity-based unplugged coding during the preschool period. *International Journal of Technology and Design Education*, *32*(1), 149–165.

Ministry of Education (2017). Digital Technologies and the national curriculum – what's it all about? Retrieved 5 December 2017 from http://elearning.tki.org.nz/Teaching/Curriculum-areas/Digital-Technologies-in-the-curriculum

Montuori, C., Gambarota, F., Altoé, G., & Arfé, B. (2024). The cognitive effects of computational thinking: A systematic review and meta-analytic study. *Computers & Education*, *210*, 104961. https://doi.org/10.1016/j.compedu.2023.104961

Rao, T. S. S., & Bhagat, K. K. (2024). Computational thinking for the digital age: a systematic review of tools, pedagogical strategies, and assessment practices. *Educational Technology Research and Development*, *72*(4), 1893–1924. https://doi.org/10.1007/s11423-024-10364-y

Rehmat, A. P., Ehsan, H., & Cardella, M. E. (2020). Instructional strategies to promote computational thinking for young learners. *Journal of Digital Learning in Teacher Education*, *36*(1), 46–62. https://doi.org/10.1080/21532974.2019.1693942

Snape, P., & Fox-Turnbull, W. (2013). Perspectives of authenticity: implementation in technology education. *International Journal of Technology and Design Education*, *23*(1), 51–68. https://doi.org/10.1007/s10798-011-9168-2

Wang, J., Fan, W., & Yu, J. (2025). Comparing the effects of unplugged activities and plugged activities on the development of students'computational thinking: a meta-analysis. *Educational Technology Research and Development*, *73*(5), 3247–3267. https://doi.org/10.1007/s11423-025-10522-w

Warren, C. A. (2018). Empathy, teacher dispositions, and preparation for culturally responsive pedagogy. *Journal of Teacher Education*, *69*(2), 169–183.

Webb, M., Davis, N., Bell, T., Katz, Y. J., Reynolds, N., Chambers, D. P., & Syslo, M. M. (2017). Computer science in K-12 school curriculua of 21 century: why, what and when? *Educational Information Technology*, *22*, 445–468. https://doi.org/10.1007/s10639-016-9493-x

Wilkerson-Jerde, M. H. (2014). Construction, categorization, and consensus: student generated computational artifacts as a context for disciplinary reflection. *Educational Technology Research and Development*, *62*(1), 99–121. https://doi.org/10.1007/s11423-013-9327-0

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. Primary and secondary computing education

Yadav, A., Hong, H., & Stephenson, C. (2016). Computational thinking for all: Pedagogical approaches to embedding 21st century problem solving in K-12 classrooms. *TechTrends*, *60*, 565–568.

Yeni, S., Grgurina, N., Hermans, F., Tolboom, J., & Barendsen, E. (2021). Exploring Teachers' PCK for Computational Thinking in Context Proceedings of the 16th Workshop in Primary and Secondary Computing Education, Virtual Event, Germany. https://doi.org/10.1145/3481312.3481320

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.