The 3C Model for Teaching Coding and Computational Thinking with an M in STEM Focus

David Martin, School of Education, Edith Cowan University, Australia da.martin@ecu.edu.au

Peter Curtis, School of Education, University of Southern Queensland, Australia Peter.Curtis@unisq.edu.au

Petrea Redmond, School of Education, University of Southern Queensland, Australia
Petrea.Redmond@unisq.edu.au

Matthew Byrne, School of Education, Edith Cowan University, Australia m.byrne@ecu.edu.au

Abstract: Australia aims to improve teacher capacity in STEM (Science, Technology, Engineering and Mathematics) education in primary school classrooms. For primary school teachers, such an endeavor requires an integrated approach when planning STEM activities. With a designated STEM curriculum, teachers could more effectively plan activities that integrate STEM subjects, particularly the capacity to incorporate mathematics and digital technologies. This paper presents a new conceptual model for planning and teaching computational thinking and coding based on existing theoretical models for teaching mathematics. Also presented is a practical example of the 3C Model's (Context, Capability, Computations) application in the context of a primary school teaching sequence that integrates Mathematics and Digital Technologies curricula using the BBC micro:bit.

Keywords: 3C Model, coding instruction, computational thinking, M in STEM, digital technologies, mathematics

Background

STEM education involves a cross-disciplinary approach designed to enhance the application of students' mathematical and scientific literacy, design and computational thinking, critical and creative thinking, and problem-solving and collaboration skills (Australian Curriculum Assessment and Reporting Authority [ACARA] 2023a). Developing these skills in students creates self-directed learners who can confidently solve real-world problems and global citizens who possess the skills essential for all 21st-century occupations (Education Council, 2015). A key intention of the Digital Technologies curriculum is to engage students in creating digital solutions to authentic problems through computer coding and computational thinking. ACARA (2023a) defines computational thinking as a problem-solving method involving several techniques and strategies that digital systems can implement. The Technologies curriculum provides a set of processes and production skills to support this process with the intention that developing students' coding and computational thinking skills be taught in an authentic context (ACARA, 2023b). But without an existing pedagogical approach and appropriate activity sequencing model, many teachers feel unprepared to plan and teach computational thinking and coding lessons using their students' interests, experiences and agency (Sentence et al., 2017). This results in a piecemeal, step-by-step approach where students reproduce without purpose akin to teaching students how to solve equations procedurally without real-world application; and undermines the Digital Technologies curriculum's rationale in developing STEM competencies, which enables students, through a deep engagement with coding, to solve real-world problems and shape their preferred future (ACARA, 2023a).

Applying mathematics in STEM is key to developing students' capacity to interpret and use their mathematical knowledge and skills in a variety of real-world situations (ACARA, 2023a). Consequently, "while it is widely acknowledged that mathematics underpins all other STEM disciplines, there is clear evidence that it plays an understated role in integrated STEM education" (Maass et al., 2019, p. 869). Version 9 of the Australian Mathematics curriculum asserts the purposeful use of digital tools and the explicit reference to the functionality of the tools. For example, in Year 2, students are to use digital tools to create picture graphs representing a data set. In Year 4, they recognize that spreadsheets, such as Excel, use a grid referencing system. In Year 6, students use dynamic geometric software to experiment with transformations (ACARA, 2023b).

There has been a call for a renewed focus on "STEM 'building blocks', especially mathematics" (Education Council, 2015, p. 5). Mathematics is an enabler (Beswick & Fraser, 2019), providing students with opportunities to use their mathematical knowledge and skills in various STEM projects. For this to occur, teachers must possess mathematics and technological pedagogical content knowledge (Koehler & Mishra, 2009). For example, if a teacher does not understand mathematical patterns and how they are constructed and developed. They will unlikely have the required knowledge to integrate coding with mathematics, where patterns become the object of learning (Palmér, 2023). Teachers are challenged to find the 'M in STEM'. Increasing teacher capacity and STEM teaching quality is a priority in Australia's National STEM School Education Strategy (Education Council, 2015).

This paper presents the background to the problem and a new conceptual model, termed the 3C Model (Context, Capability, Computations) (Martin et al., 2024), that provides a pedagogical approach for integrating mathematics into coding lessons based on theoretical models. Also presented is an example of a teaching sequence that authentically integrates mathematics with coding using the BBC micro:bit and describes its viable pedagogical approach.

Cognitive development as a theoretical basis for the 3C Model

Piaget's (1964) theory of cognitive development delineates four developmental stages through which children progress in their ability to understand and process information: The Sensorimotor Stage (Birth to 2 years), the Preoperational Stage (2 to 7 years), the Concrete Operational Stage (7 to 11 years), and the Formal Operational Stage (11 years and older). Primary school students would be performing mostly at the concrete operational stage. While these students can think abstractly to some extent, Piaget (1964) emphasized the importance of concrete experiences in students' learning as they help them grasp the underlying principles, concepts, and skills before moving to their abstract representations. Drawing from Piaget's (1964) cognitive development theory, the 3C Model (Martin et al., 2024) was designed to teach coding and computational thinking concepts and skills to primary school children (ages 7-11) who are performing primarily at the concrete operational stage. Pedagogically, the 3C Model is based on instructional models used for teaching mathematics to primary school children which leverage emerging abstract thinking by incorporating concrete materials, physical movement, age-appropriate language representations, and authentic experiences. The purposeful design of the 3C Model makes accessible to teachers an organized and effective method for teaching the concepts and skills related to mathematics, coding, and computational thinking.

The 3C Model: A New Conceptual Model Based on the Concrete-Representational-Abstract (CRA) and Language Models

Conceptual models illustrate the interaction between key concepts or variables; they organize ideas, clarify concepts and provide representations for interpreting knowledge. It is a way to unpack complex theoretical ideas that might explain, predict, and understand phenomena (Abend, 2008). The conceptualization of the 3C Model (Martin et al., 2024) based foremost on the work of Seymour Papert's (1980) who coined the term "body syntonic" to relate a student's perception of body movements to computational thinking (p. 63). His vision was that students' programming activities would lead to a new learning approach and a consequential transformation of pedagogies concerning subjects fundamental to the current STEM construct. Two issues block the development of Papert's vision: the availability of a coherent pedagogical approach and an appropriate activity sequencing model (Sentence et al., 2017). The teaching of coding and computational thinking is commonly presented to children in a procedural and decontextualized manner because there is no pedagogical best practice available to teachers (Sentence et al., 2017). These issues were the genesis of the 3C Model which integrates contextual problems, device capabilities and coding concepts through connection with the concrete-representational-abstract (CRA) (Jordan, 1998; PaTTAN, 2020; Peterson et al., 1988) and Language Model (Irons & Irons, 1989; Irons, 2014). Both are pedagogical approaches commonly used for teaching mathematics and provide the conceptual and pedagogical foundation of the 3C Model.

The Concrete-Representational-Abstract (CRA) Model for Teaching Mathematics

The CRA Model (Jordan, 1998) is an approach to teaching mathematics that combines effective components of direct and explicit instruction and constructivist practices. The CRA sequence of instruction begins at the concrete level, which involves students learning using concrete materials, often called manipulatives. For example, using geometric solids promotes a conceptual understanding of 3D objects. The teacher starts the lesson by explicitly demonstrating how to represent and solve math problems with concrete materials that relate to the concept or skill being taught.

Following explicit modeling, students engage with the manipulatives and model the concepts or skills, practicing what the teacher modeled while receiving feedback and praise. When students achieve mastery related to solving problems at the concrete level, the sequence of instruction progresses to the representational level. This stage of the teaching sequence involves activities that use semi-concrete materials, such as graphics or pictures, that address the same concepts or skills learned previously. The lessons utilize the gradual release model (Pearson & Gallagher, 1983). Once mastery is achieved at the representational level, the instruction sequence progresses to the abstract level. The activities at this stage involve students solving similar problems using numbers and symbols. For students who initially struggle with the abstract background, the teacher can differentiate their learning by allowing them to concurrently use concrete or semi-concrete materials until they can independently model the concept or skill symbolically.

The Language Model for Teaching Mathematics

The Language Model (Irons & Irons, 1989is also an effective pedagogical approach to teaching mathematics. It stems from the work of Payne and Rathmell (1975) who proposed a teaching guide using a triangular structure (known as Rathmell's Triangle) for developing mathematics concepts (Figure 1). The arrows that form the triangle establish the basis of the teaching approach, which is to prompt students to develop relationships between concrete and graphic representations of a number's value, the number itself and the number expressed in words.

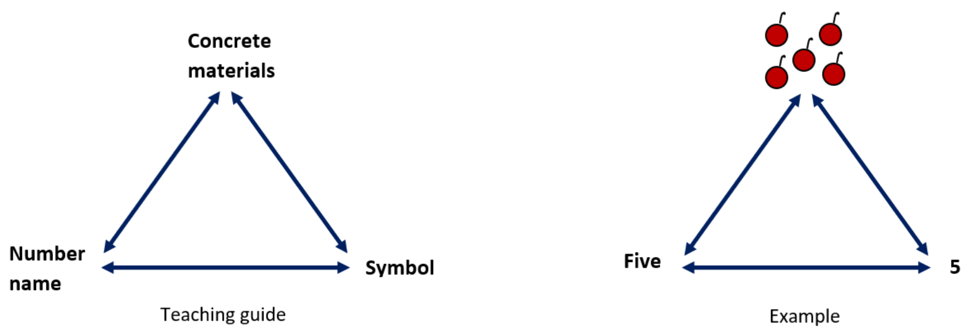

Figure 1. Rathmell's triangle for teaching mathematics concepts (modified from Payne and Rathmell, 1975)

The Language Model (Irons & Irons, 1989) extends the CRA Model (Jordan, 1998) by considering the explicit language teachers and students should use at each learning stage. Since mathematics has a language of its own, this makes sense. In the Language Model context, 'language' represents student and teacher dialogue surrounding mathematical concepts and skills. The teacher initiates the learning process by modeling concepts and skills using authentic concrete materials alongside specific language relative to the stage of learning. Next, the students use the same language and concrete materials to develop their knowledge and understanding of concepts and skills. Again, this approach makes sense since "Mathematical concepts are not developed in the absence of mathematical language" (Australian Education Council, 1991, p. 20). As a learning strategy, students should be given opportunities to practice the language as they share strategies and solutions to problems. Students are more likely to learn concepts and skills when they have well-defined approaches to describing and discussing their experiences (Australian Education Council, 1991). Figure 2 illustrates the Language Model.

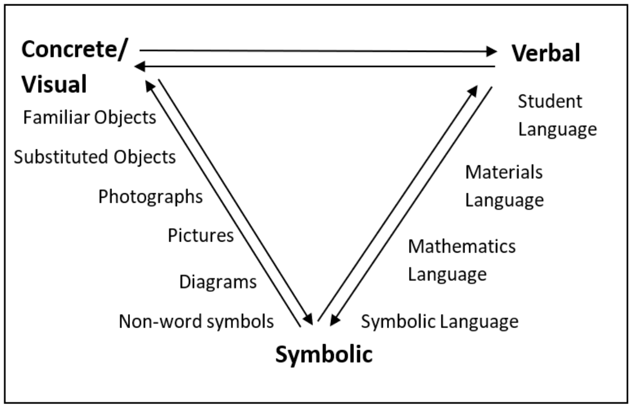

Figure 2. The Language Model for mathematics teaching and learning (Irons, 2014).

Discussion

The 3C Model for Teaching Coding and Computational Thinking

The 3C Model's teaching approach provides a similar 'learning journey' as the CRA and Language Models through connection with their theoretical and pedagogical approaches as students link initial physical movement and everyday language to the abstract notion of coding and computational thinking. This alternative activity sequencing model, specifically designed to create opportunities for using coding and computational thinking with physical devices within a STEM-focused school, was developed and focused on three stages: Context of the problem, Capabilities of the device, and Computational concept of the unit of work.

Putting the 3C Model into practice

The three key elements of the 3C Model are embedded in the first three lessons of a 5-lesson teaching sequence. Figure 3 illustrates the teaching sequence underpinned by the CRA and Language Models.

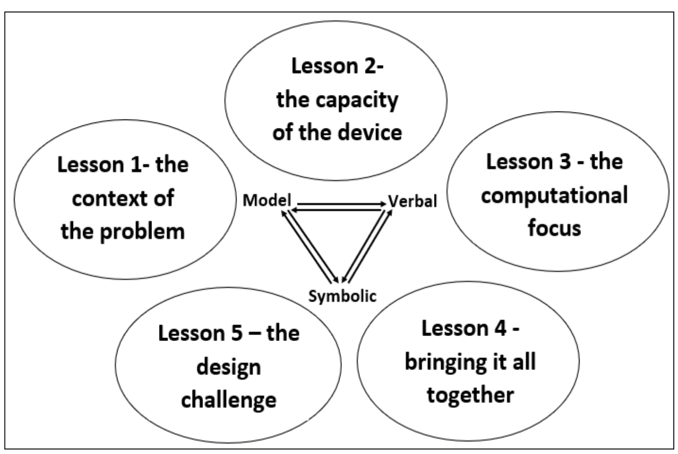

Figure 3. A pedagogical approach using the 3C Model (Martin et al., 2024)

Lesson 1- *Context*: To have mathematics make sense, or any other curriculum content, the unit of work needs to be in the context of an authentic problem. These problems can be sourced from the curriculum, such as Science or the Humanities, or the children's experiences. In either case, the students learn curriculum content to solve the chosen problem, which is not the same one they will solve for themselves later in the lesson sequence. In this sense, the lesson sequence commences with the child and the authentic context at the center.

Lesson 2 addresses *Capabilities*. In this lesson, students learn about the capabilities of the device. In other words, what can the tools and features of a digital device or software accomplish? The pedagogical approach to lesson 2 is for the teacher to show the students a pre-coded program results rather than the code. During this juncture of the lesson, the teacher asks the students to explain their observations and articulate what they observe in everyday language. Students do not code at this point. Instead, they are asked to make statements that are considered pre-cursers to code, for example, If … then statements. This sets the groundwork for establishing the sentence stems and language patterns necessary for linking the symbolic code. The strategy underpinning this stage of learning is comparable to the first stage of the CRA and Language Models, where the teacher makes the critical connection between a visual or physical representation and a common language.

Lesson 3 addresses *Computational thinking,* which is sourced from the curriculum. For example, the Year 1 and Year 2 Digital Technologies curriculum requires students to: follow and describe algorithms involving a sequence of steps, branching (decisions) and iteration (repetition) (ACARA, 2023a). While there is a Digital Technologies curriculum focus for lesson 3, there is equally a focus on getting the students to act out the algorithms they learned about in lesson 2 (for example, acting out the If … then statements). Lesson 3 is comparable to the concrete stage of the CRA and Language Models. The teacher models a sequence of coded instructions used in lesson 2 and asks the students to use concrete materials and movement to act out the sequence of instructions while saying the words. During these interactions, a student-recorder writes down these movements as step-by-step algorithms using their language or by drawing flowcharts. This lesson activity is consistent with the representational stage of the CRA and Language Models, where students record and articulate a concept or skill in a familiar format. The students need to use this pseudo-code until they have sufficient experience and conceptual understanding before introducing symbolic code.

In lesson 4, the students link the modeling, the familiar language, the conceptual vocabulary and the symbolic code. The aim is for the students to connect the writing of a sequence of instructions, or the drawing of flowcharts, to the computer code to demonstrate their understanding of the device's inputs, processes, and outputs. This is comparable to the abstract phase of the CRA and Language Models when the teacher shows students how to solve mathematics problems symbolically. Lesson 4 also provides an opportunity for students to realize why they needed to learn, for example, the mathematics content from lesson 1 because it is embedded in what will be coded to solve the new problem. The teacher should be prepared to revisit visual or physical representations or connect to familiar language with students struggling with the symbolic algorithms in this lesson. This scaffolding mechanism brings the student back to the *pseudo code* or acting out of the computer-coded algorithms as the representational phases of the learning.

In lesson 5, the teacher sets up the environment for students to decide on a problem they may care to solve. This may be a time to provide the students with a choice of contexts and problems. There is no more explicit or guided teaching. The students now have seen the code developed and consequently have a conceptual understanding of the code and can transfer it to a new context. Thus, the teacher hands over responsibility to the students to solve a new but similar problem, with the teacher acting as facilitator and coach. In other words, this lesson is where a student-centered, inquiry-based approach begins for the students and the teacher. Lesson 5 is also where the Technologies curriculum's processes, production skills, and design process are in play, depending on the product or service being created (ACARA, 2023a). Hence, in most cases, lesson 5 starts a Design and Technologies challenge that may go on for two or more lessons. Table 1 provides an example of the 3C activity sequence model in practice.

Table 1. Example of a five-lesson teaching sequence using the 3C Model and micro:bits

Unit Summary: Year 4 Sun Safety and Micro:bits Students will investigate the health issue of sun exposure, its effect on skin damage, and the potential development of skin cancer as a contextual problem. Students will design a solution to the problem that will assist them in monitoring and moderating the time they are in the sun. Students will be introduced to the idea of algorithms and branching statements through experimentation of the capabilities of the micro:bit device, with its capacity to measure time, countdown display and trigger events. Students will engage with mathematics in terms of working with time conversion issues and the development of an algorithm. Students will engage with the following curriculum elements:

Digital Technologies - Processes & Production (P&P) Skills

- Define simple problems, describe and follow a sequence of steps and decisions (algorithms) to solve them.
- Implement simple digital solutions as visual programs with algorithms involving branching (decisions) and user input.

Mathematics

- Solve problems involving the duration of time, including situations involving “am” and “pm” and conversions between units of time.
- Create and use algorithms involving a sequence of steps and decisions and digital tools to experiment with identifying, interpreting, and describing emerging patterns.

Design and Technology

- Recognize the role of people in design and technology occupations and explore factors, including sustainability, that impact the design of products, services, and environments to meet community needs.
- Plan a sequence of production steps when making designed solutions individually and collaboratively.

Health

- Identify and practice strategies to promote health, safety, and well-being.

Lesson Sequence	**Objective(s):**	**Curriculum Connections**
Lesson 1: Learn curriculum content through *Context*. Students are introduced to the context of the health content description through a series of activities that normally constitute a health lesson. These might include health activities such as recording personal sun exposure duration. The mathematical activities comprise calculating the time spent in the sun and revising a.m. and p.m. terminology and concepts.	Students will: ● recognize the dangers of sun exposure and make the connection between sun exposure and the development of skin cancer. ● calculate the duration of time using a.m. and p.m. notation. ● convert units of time using relationships between 60 seconds in a minute and 60 minutes in an hour.	Health Mathematics
Lesson 2: Learn the *Capabilities* of the digital device. Students work with pre-coded micro:bits organized by the teacher. At this point, students are not exposed to the actual code. The micro:bits provided by the teacher (consider group rotations) will comprise two categories: 1. Micro:bits that measure and respond to light levels in a variety of ways, e.g., shining a torch on the micro:bit, it plays a sound or audio byte, and displays a text message or number on the screen. 2. Micro:bits that measure a lapse of time, e.g., a text message, number, or audio alert, which is given after a countdown is displayed on the LED display. This lesson assists students to: ● determine the differences between input (time, light levels) and output (text or numerals on the LED, a triggered sound or audio file).	Students will: ● determine through observation and experimentation that the micro:bit can respond to variables such as the passage of time and levels of light. ● develop everyday coding language. ● follow and analyze algorithms involving a sequence of steps and branching decisions that use addition and identify	Digital Technologies P&P skills Mathematics

● develop everyday coding language such as "when and then…" e.g., "When the countdown is finished, the micro:bit tells me that time is up" ● inductively generate statements such as: "When I shine a torch on the micro:bit, then it displays the number that shows the strength of the light "	and describe emerging patterns.	
Lesson 3: Learn *Computational Thinking* Students use movement (body syntonic) to act out the action of the algorithm. This is done with a design idea from the teacher, e.g., the teacher creates a program for a bike light that is used at night and relies on the micro:bit to determine that the light level has dropped too low. After the role-play, the algorithm is expressed in everyday language (sentences), typically on hard paper or cardboard strips. Mathematically, this may involve the teacher using direct instruction, showing the application of calculation skills, e.g., the conversion of seconds to minutes, and the design of the algorithm. This is again recorded in everyday language.	Students will: ● follow and act out simple algorithms using coding vocabulary. ● record the coding sequences in English, pseudocode or as flowcharts.	Mathematics Digital Technologies P&P skills
Lesson 4: Bringing lessons 1-3 together. The teacher leads the students to draw connections between the actions in lesson 3, written using everyday language that represents the action and the coding blocks. As a teaching strategy, consider populating on hard paper or cardboard strips with abstract code sets representing the algorithms. These strips are then arranged side by side with the everyday sentences (written in lesson 3) so that the connections are supported and obvious. Consider group rotations again for each set of abstract codes representing the algorithms arranged alongside the everyday sentences written in lesson 3.	Students will: ● draw connections between mathematical formulas and contextualized statements to abstract code.	Mathematics & Digital Technologies P&P skills
Lesson 5: The design challenge begins Some students revisited lesson 1 (in many cases, the health lessons have been taught in parallel throughout the lesson sequence). Students brainstorm situations where they may need to monitor their sun exposure. Examples include: ● on the sporting field, at the beach, on a boat. during lunchtime Many other students generalized the micro:bit's ability to respond to changes in light levels and creatively applied them to new situations. For example, a pair of students designed a watering device that switched on when light levels dropped using the If, Then statement. In each design, students begin to collaboratively design, code and calculate. Differentiation can occur by returning to the everyday language or the roleplay phase.	Students will: ● design and implement a digital solution using MakeCode for micro:bits with algorithms involving a sequence of steps, branching decisions, and user input. ● calculate the duration of time using a.m. and p.m. notation. ● convert units of time using relationships between 60 seconds in a minute and 60 minutes in an hour. ● apply practices that support collaborative	Digital Technologies P&P skills Mathematics Design & Technology

	problem-solving.	

This instructional sequence was tested with Year 6 students and the results will be reported in full in another paper. The research used semi-structured interviews in group settings to explore and examine the students' lived experiences while engaged in their learning under the 3C Model. Five themes emerged from the analysis of the data: 1) Opportunity to creatively develop multiple pathways to solutions, 2) Availability of varied resources to discover multiple pathways to solutions, 3) Positive challenge and the reality of perseverance, 4) The importance of visuals, and 5) The connection between language and code. The students' perspectives reflected their learning progressions of acting out algorithms and computational thinking using movement and concrete materials, followed by engaging with visual representations and corresponding abstract language.

Theme 1 indicated their recognition of the opportunity to creatively develop multiple pathways to their solutions. For example, *You have to find out something, but [in Mathematics] there is still 'the answer'. In coding, you are working on different things and you have to find a way to make your coding work.* This indicative response supports the aim of the Australian Digital Technologies curriculum of developing in students the ability to use computational thinking skills creatively when formulating digital solutions to authentic problems (ACARA, 2023a).

Within theme 2, student responses reflected engagement with diverse resources, including their peers, to exchange ideas and explore solutions within a socially constructed learning environment. One student stated: *Different people have different things to offer too - for instance, (student name) helped a lot with the platform building because he's good with building things. He was also good when it came to solving problems as well*. Another student echoed this sentiment stating, *You get the chance to ask other people and you learn from them as well*. The collective interpretation of responses in theme 2 aligns with Bers et al. (2019) findings, highlighting the significance of socially constructed knowledge as an integral aspect of the learning process.

Comments from students reflecting on theme 3 conveyed their positive experiences with authentic challenges and the importance of perseverance when facing these challenges, *The stuff, it's a lot harder but this is more friendly. But with coding, you can just keep working and it's really exciting when you can solve that problem.* Another student stated, *I think getting into trouble with your work [algorithms not working as expected] helps you ... I think it expands your knowledge.* The views expressed by the students are supported by Sirakaya et al. (2020), indicating that a strong predictor of positive attitudes in STEM was associated with students' preference towards authentic learning.

The fourth and fifth themes relate to Piaget's (1964) levels of cognitive development and the Language Model (Irons, 2014) in that there is a progression from the concrete operational stage (acting out the computational thinking using movement and concrete materials and visual representations) to the abstract (in the form of code). When asked about the importance of visuals (theme 4), a student conveyed, *I think this way is good because you can ask questions and you can use different ways like drawing it to show you what you actually can do. If you are in a book, then you don't get to visually see what it does. In a book you have to think, you can't see it.*

For the last theme, code is apparent in the language, a student who created a scanning device using touch and light sensors to help blind people recognize products in stores provided the following: *When you touch the touch sensor, it starts the color sensor, and if the sensor saw a particular color, then it was [acted] like a trigger to play an audio file - like if it's all orange, I programmed it to play the audio file; this is a can of beans, but if it saw blue, then would say 'this is a loaf of bread'.*

This paper introduced the 3C Model and provided an example of how primary school teachers could effectively utilize the pedagogical sequence in STEM education. The 3C Model allows primary school teachers to build both coding and computational thinking skills in students authentically and sequentially. The 3C Model's teaching sequence can be applied to the CRA (Jordan, 1998) and Language Model's (Irons & Irons, 1989) pedagogical approach to teaching mathematics. Like programming languages, mathematics has its own language (Larkin et al., 2012). As best practice dictates, abstract concepts and associated skills are represented initially using concrete materials and physical movement while describing experiences using everyday language. To scaffold the learning, the teacher switches to using semi-concrete materials such as drawings and pictures such as using circles and tallies to imprint a picture of the concept and skill while modeling more formal mathematics language. Lastly, at the symbolic stage, the teacher models the same concept and skill on a whiteboard using symbols and numbers while continuing to develop the students' language. Applying the CRA and Language Model to the 3C Model involves initially using two-way interactions between visual and physical movement with concrete materials and everyday language representations. The sequence should then build on the same concepts and skills using three-way interactions between semi-concrete materials, such as pseudo-code or flowcharting, related vocabulary, and symbolic algorithms.

Without a designated STEM curriculum, primary school teachers face challenges when planning activities that integrate STEM subjects (Kurup et al., 2019), particularly the capacity to integrate mathematics, the subject acknowledged as underpinning all STEM learning (Goos et al., 2023). Developing STEM competencies in teachers enables their students to effectively develop, model, analyze and improve solutions to real-world problems. These STEM competencies are required to fill the deficit in current and forecasted future STEM workforces (Office of the Chief Scientist, 2021). Teachers can utilize the 3C Model to develop these required STEM competencies.

As a pedagogical approach, the 3C Model is versatile as it can be used in other curriculum planning. It can afford a consistent approach in terms of embedding the needed coding and computational thinking skills to solve real-world problems across the curriculum. As highlighted by Maass et al. (2019), the role of mathematics in interdisciplinary STEM education is often underrepresented, and students may need access to the needed computational thinking skills and mathematical understanding to be confident learners. Using the 3C Model in a primary setting enables further opportunities for mathematical understanding to be developed across the curriculum and through STEM learning.

Implications and Limitations

There are implications for practice when using the 3C Model. Firstly, it provides a coherent pedagogy and an appropriate activity sequencing model for teachers that is embedded in the literature and common practice for the teaching of mathematics. Secondly, the 3C Model addresses a key consideration of computational thinking in the Mathematics curriculum in that students are to develop the capacity to purposefully select and effectively use the functionality of a digital device, platform, software, or digital resource. Thirdly, the model provides teachers and students with mathematical and computational metalanguage and a common understanding of teaching coding.

A limitation of this study is that the 3C Model has been tested with just Year 6 students within a coding club at one school using qualitative methods. Future research will test the model using additional data collection methods and varied types of practical examples with pre-service teachers, practicing teachers and primary school students at various year levels. In this way, the 3C Model needs further validation, particularly with classroom teachers.

Conclusion

This paper applies two theoretical constructs in the development of a new conceptual model for teaching coding and computational thinking with a focus on mathematics. It provides a practical example of how the 3C Model works in practice. While learning through the 3C Model instructional process, students respond to questions and make meaningful interpretations while developing their coding and computational thinking skills. They then plan and conduct investigations, make observations, take measurements, and process and analyze data and information while evaluating and communicating. These are required skills that lay the foundation of future STEM workforces.

The versatility of the 3C Model is a direct response to the National STEM strategy and highlights the needed collaboration of drawing together curriculum documents such as Mathematics and Technologies curricula. The use of the model can be a way to proactively address the underrepresentation of the role of mathematics in interdisciplinary STEM education (Maass et al., 2019). The 3C Model has further utility in that its pedagogical approach can be used by teachers across varied curricula as well as in STEM education. In the context of STEM skill requirements for the future workforce, the 3C Model embeds coding and computational thinking skills students need to help solve real-world problems and build their capacity to shape their preferred futures.

References

Abend, G. (2008). The meaning of theory. *Sociological Theory, 26*(2), 173-199. https://doi.org/10.1111/j.1467-9558.2008.00324.x

Australian Curriculum Assessment and Reporting Authority [ACARA]. (2023a). *Version 9 Australian Curriculum: Technologies.* https://v9.australiancurriculum.edu.au/teacher-resources/understand-this-learning-area/technologies#technologies

Australian Curriculum Assessment and Reporting Authority [ACARA]. (2023b). *Version 9 Australian Curriculum: Mathematics.* https://v9.australiancurriculum.edu.au/f-10-curriculum/learning-areas/mathematics/years-1_2_3_4_5_6

Australian Education Council. (1991). Enhancing mathematics learning. In *A National Statement on Mathematics for Australian Schools* (pp. 16-24). Curriculum Corporation. https://eric.ed.gov/?id=ED428947

Bers, M. U., González-González, C., & Armas–Torres, M. B. (2019). Coding as a playground: Promoting positive learning experiences in childhood classrooms. *Computers & Education, 138*, 130-145. https://doi.org/10.1016/j.compedu.2019.04.013

Beswick, K., & Fraser, S. (2019). Developing mathematics teachers' 21st century competence for teaching in STEM contexts. *ZDM - Mathematics Education*, *51*(6), 955-965. https://doi.org/10.1007/s11858-019-01084-2

Education Council. (2015). *National STEM School Education Strategy, 2016–2026*. https://www.education.gov.au/australian-curriculum/support-science-technology-engineering-and-mathematics-stem/national-stem-school-education-strategy-2016-2026

Education Services Australia. (n.d.). *Digital Technologies Hub: Core concepts* https://www.digitaltechnologieshub.edu.au/understanding-dt/the-dt-curriculum/core-concepts/

Goos, M., Carreira, S., & Namukasa, I. K. (2023). Mathematics and interdisciplinary STEM education: recent developments and future directions. *ZDM–Mathematics Education, 55,* 1199-1217. https://doi.org/10.1007/s11858-023-01533-z

Irons, C., & Irons, R. (1989). Language experiences: A base for problem solving. In P. Trafton & A. Shulte (Eds.), *1989 Handbook: New Directions for Elementary School Mathematics*. National Council of Teachers of Mathematics. (pp. 85-98). https://archive.org/details/newdirectionsfor0000unse_i4w2/mode/2up

Irons, R. (2014). Language is the core for the concept of addition. *Educating Young Children: Learning and Teaching in the Early Childhood Years*, *20*(1), 38-41. https://ecta.org.au/wp-content/uploads/2019/06/Educating_Young_Children_Journal-20.1-2014.pdf#page=40

Jordan, L., Miller, M.D., & Mercer, C.D. (1998). The effects of concrete to semi-concrete to abstract instruction in the acquisition and retention of fraction concepts and skills. *Learning Disabilities: A Multidisciplinary Journal*, *9*(3), 115-122. https://js.sagamorepub.com/index.php/ldmj/article/view/5583

Koehler, M., & Mishra, P. (2009). What is technological pedagogical content knowledge (TPACK)? *Contemporary Issues in Technology and Teacher Education*, *9*(1), 60-70. https://www.learntechlib.org/primary/p/29544/.

Kurup, P. M., Li, X., Powell, G., & Brown, M. (2019). Building future primary teachers' capacity in STEM: based on a platform of beliefs, understandings and intentions. International Journal of STEM Education, 6, Article 10. https://doi.org/10.1186/s40594-019-0164-5

Larkin, K., Jamieson-Proctor, R., & Finger, G. (2012). TPACK and pre-service teacher mathematics education: Defining a signature pedagogy for mathematics education using ICT and based on the metaphor "mathematics is a language". *Computers in the Schools, 29(*1–2), 207-226. https://doi.org/10.1080/07380569.2012.651424

Maass, K., Geiger, V., Ariza, M. R., & Goos, M. (2019). The role of mathematics in interdisciplinary STEM education. *ZDM - Mathematics Education*, *51*, 869-884. https://doi.org/10.1007/s11858-019-01100-5

Martin, D. A., Curtis, P., & Redmond, P. (2024). Primary school students' perceptions and developed artefacts and language from learning coding and computational thinking using the 3C model. *Journal of Computer Assisted Learning*, Advanced online publication. https://doi.org/10.1111/jcal.12972

Office of the Chief Scientist. (2021). 2021 Rapid report-Space industry and the STEM workforce. https://www.chiefscientist.gov.au/news-and-media/2021-rapid-report-space-industry-and-stem-workforce

Palmér, H. (2023). Children (Aged 3–5 Years) Learning Mathematics Through Programming, Thinking and Doing, or Just Doing? In *Teaching Coding in K-12 Schools: Research and Application,* (pp. 13-22). Springer. https://doi.org/10.1007/978-3-031-21970-2_2

Papert, S. (1980). *Mindstorms: Children, Computers, and Powerful Ideas*. Basic Books.

PaTTAN. (2020). *Concrete-Representational-Abstract: Instructional Sequence for Mathematics*. https://www.pattan.net/getmedia/9059e5f0-7edc-4391-8c8e-ebaf8c3c95d6/CRA_Methods0117

Payne, J. N., & Rathmell, E. C. (1975). Mathematics Learning in Early Childhood: Number and Numeration. *National Council of Teachers of Mathematics Yearbook, 37,* 125-160. https://eric.ed.gov/?id=EJ121631

Pearson, P. D., & Gallagher, M. C. (1983). The instruction of reading comprehension. *Contemporary Educational Psychology, 8(*3), 317-344. https://doi.org/10.1016/0361-476X(83)90019-X

Peterson, S. K., Mercer, C. D., & O'Shea, L. (1988). Teaching learning disabled students place value using the concrete to abstract sequence. *Learning Disabilities Research*, *4*, 52-56. https://psycnet.apa.org/record/1989-38227-001

Piaget, J. (1964), Part I: Cognitive development in children: Piaget development and learning. *Journal of Research in Science Teaching, 2*(3), 176-186. https://doi.org/10.1002/tea.3660020306

Sentance, S., Waite, J., Yeomans, L., & MacLeod, E. (2017). Teaching with physical computing devices: the BBC micro: bit initiative. In E. Barendsen & P. Hubwieser (Eds.), *Proceedings of the 12th Workshop on Primary and Secondary Computing Education,* (pp. 87–96). The Association for Computing Machinery. https://doi.org/10.1145/3137065.3137083

Sirakaya, M., Alsancak Sırakaya, D., & Korkmaz, Ö. (2020). The impact of STEM attitude and thinking style on computational thinking determined via structural equation modeling. *Journal of Science Education and Technology, 29*, 561-572. https://doi.org/10.1007/s10956-020-09836-6