Research Paper in support of the introduction of Technology

in a redeveloped Primary School Curriculum

Digital Technology – Using, Understanding and Creating

Jane Waite
Queen Mary University of London

Keith Quille
Technology University Dublin

Contents

1. Introduction

The *Draft Primary Curriculum Framework* (NCCA 2020), published for consultation, proposes the inclusion of a number of new areas of learning, one of which is the introduction of Digital Technology sitting within the curriculum area of 'Science and Technology'. The draft framework presents a description of learning in relation to digital technology which intends to support children as "creative, confident and critical users of digital technology" (NCCA 2020, p. 8). In support of this aim and the development of associated curriculum provision, a suite of research papers has been developed.

Included in this set of papers, is an extensive technical report related to the 'Using, Understanding and Creating' aspect of digital technology. The technical report synthesises earlier NCCA studies and reports, as well as national and international research in computing education. Through the synthesis of research, the technical report presents a model of curriculum design to support learning in digital technology. The technical report also provides examples of how to apply the model as well as pedagogical guidance for consideration in a redeveloped Primary School Curriculum. This paper is an abridged version of the technical report.

2. Rationale

Increasingly, digital technology is seen across the world as an important aspect of learning from early childhood to adult education. This paper focuses on “Using, Understanding and Creating” with digital technology which has been highlighted nationally and internationally as an important aspect of learning in this area. For example, the EU Digital Action Plan, specifically, “Strategic Priority 2, Enhancing digital skills and competences for the digital transformation” (European Commission 2020, p. 12), discusses several factors that are relevant in informing policy in the area of using, understanding and creating with digital technology:

- A changing society and the transition to a green and digital economy require solid digital competences.
- Digital literacy has become essential for everyday life.
- Computing education in schools allows young people to gain a sound understanding of the digital world.
- To thrive in a technology-driven economy, Europeans need digital skills.
- Advanced digital skills are in high demand.
- All Member States face shortages of digital experts.
- Women accounted for 54% of all tertiary children in the EU in 2017, yet they are particularly underrepresented in the digital sectors.
- Everyone should acquire a basic understanding of new and emerging technologies including AI.

(European Commission 2020, pp. 12-14)

The EU Digital Action Plan also recommends improving the provision of digital skills in education and training, by investing:

> ....in teacher professional development, exchange of best practice on instructional methods, including through a focus on inclusive high-quality computing education

> (informatics) at all levels of education and fostering dialogue with industry on identifying and updating new and emerging skills needs, in synergy with the skills agenda.
>
> (European Commission 2020, p. 15)

Terminology related to digital technology is widely contested and can be quite fluid as generations of technologies evolve. Terms such as computer science, informatics, computing, and digital literacy are all prevalent in education. This paper uses the terms set out in the *Draft Primary Curriculum Framework* which include “Being a digital learner” as a key competency (NCCA 2020, p. 8) and “Technology Education” within the broad curriculum area of “Mathematics, Science and Technology Education” (NCCA 2020, p. 11). In particular this paper focuses on the ‘understanding, using and creating’ aspects of digital technology which resonates with all of the terms listed previously.

By understanding, using and creating with digital technology primary-aged children can develop skills that are highly transferable to any domain (Computing Crossroads 2021). It is the opinion of the authors that through understanding, using and creating with digital technology, all of the key competencies described in the *Draft Primary Curriculum Framework* can be progressed:

> “Being a digital learner, being mathematical, communicating and using language, fostering wellbeing, learning to be a learner, being an active citizen and being creative”.
>
> (NCCA p. 10, 2020)

This is a strength that can be overlooked when considering this subject for inclusion in a curriculum.

Learning associated with digital technology can be viewed as a combination of more traditional curriculum areas/subjects, including science, mathematics, and engineering (Tedre 2014). It also can involve aspects of the arts and the social sciences as learners engage in the creation of novel solutions and consider the ethical and societal impacts of their designs. Having both discipline specific learning as well as cross-curriculum application digital technology could be best located within areas such as science, mathematics, design, and technology or as a separate, distinct subject. Decisions to this effect are often based on an individual countries' historical and political context. In Ireland, it is proposed that digital technology is incorporated into the Mathematics, Science and Technology curriculum area for Stages 1 and 2, and more specifically within the subject of Science and Technology in Stages 3 and 4 (NCCA 2020, p. 11). The integration of digital technology in Stages 1 and 2 is particularly apt, as learning in these years generally focuses on the design and construction elements of digital technology; while the more conceptual understanding and mathematical aspects of this learning is revealed as learners dig deeper into the discipline over time.

3. The Concepts and Processes Model

This section presents the Concepts and Process model (CP Model) which has been established through an extensive synthesis of literature outlined in the technical report which accompanies this paper. The *CP Model* provides a structure that could be used to develop curriculum provision in digital technology. We first describe the *CP Model* overall and then outline each of the model's descriptors.

The model has two dimensions, dimension 1; 'Concepts' or knowledge and understanding (C) and Dimension 2; 'Processes' or creating and production skills (P), which combined can be thought of as the building blocks for primary school curriculum provision. The *CP Model* is based on modern curriculum approaches which support the knowledge, skills, dispositions and values associated with the foundational underpinnings of digital technology.

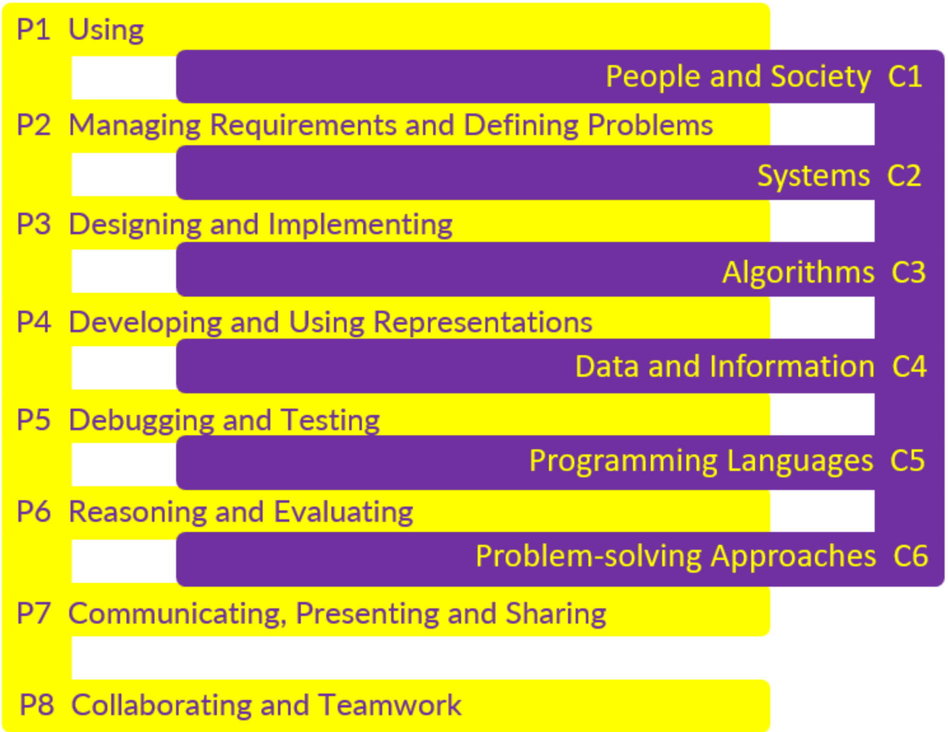
The design of the *CP Model* is a synthesis of recent computing education research and digital competency model development. The method used to develop the *CP Model* included the cross-referencing of a wide range of sources to find themes and competencies from existing research-derived computing competency models, international curricula objectives, and findings from previous NCCA studies.

Further application of the *CP Model* will need to be grounded in curriculum design processes, which should include negotiation, consultation, and deliberation. It is also important to note that, although the focus for this model was on the understanding and creating of digital technologies, a competency-based approach naturally incorporates both concepts and skills. Therefore, the presented model is likely to have further application in the context of the 'being a digital learner' competency.

4. CP Model Descriptors

Within the two dimensions of the *CP Model*, there are six *Concept* (C) or knowledge and understanding themes and eight *Process* (P) and production themes (see Figure 1) with a summary of each.

Figure 1: CP Model–Concepts and Processes

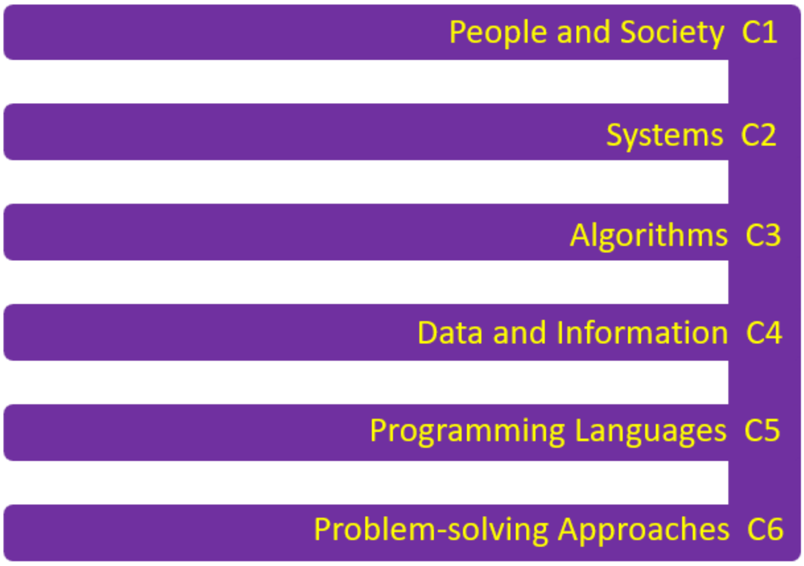


As a caveat, the following theme summaries (C1 to C6 and P1 to P8) are not exhaustive; they do not detail progression or offer guidance when the themes should be introduced. We see this as a matter for curriculum design process involving consultation, negotiation and deliberation in the Irish education context.

Concepts

There are six themes within the *Concept Dimension* as shown in Figure 2 below. The concepts incorporate literacy and cognitive competencies, intra-personal and knowledge domains.

Figure 2: Concept Dimension – Concepts or Knowledge and Understanding



C1 - People and Society

Digital technologies are not created in a vacuum; they are developed for people within a social context. Children should show a developing knowledge of digital technology's historical, current, and future uses within this concept. Children will also start to understand the impact this has on their own lives, familiar and broader groups, and society. Included in this is an appreciation of the sustainability of digital solutions and their impact on the environment and economy. Through this, children start to understand the world in general and the role of technology to meet the needs of people, and society, including preparing them to understand the ethical considerations of digital technologies ranging from a personal to a broader social viewpoint. When designing and evaluating digital technologies, they will also consider how and why they should act respectfully and responsibly online.

C2 - Systems

The concept of a system, a set of things that work together, is central to learning about digital technologies. This includes knowledge of the components of digital systems and how they interconnect as well as understanding, on a basic level, about hardware and software. Children should also learn about the purpose of everyday digital system components, including understanding about simple inputs and outputs, such as keyboards and screens. When learning about networks, children will start to understand, on a simple level, how networks transmit data. They are also likely to use, recognise, reason and evaluate, communicate and present about systems. When learning about these systems and particularly networks there may be opportunity to identify links to ethics and internet safety. There are particular opportunities to teach about components of systems when working with physical computing and or unplugged activities.

C3 – Algorithms

Children's understanding of algorithms begins with them following, or physically enacting, simple instructions, including making choices at decision points. Through this process, they start to understand the need for accuracy in putting steps in the correct order and including the right level of detail so there will always be the same outcome. This work on instructions helps children understand what an algorithm is; a precise and unambiguous set of instructions for completing a task or solving a problem. The complexity of the algorithms that children use, design, implement, test, debug and evaluate will increase through developmental stages. Young children may enact an algorithm through movement and role-play, older children may write more formal representations with lists and labelled diagrams, even with flowcharts. In primary classrooms, it is likely that block-based programming languages, such as Scratch (Massachusetts Institute of Technology 2021), or simple text languages, such as LOGO (Massachusetts Institute of Technology 2015), will be used to implement designs (including algorithms). However, it is essential to remember that algorithms can also be implemented through menu-driven software, such as when making games, animations, quizzes, or presentations.

C4- Data and Information

For young children, learning about data is essential as it increasingly underpins systems that influence their lives through machine learning products. More traditionally developed systems also use data, and primary-aged children should be introduced to the vocabulary and concepts associated with data for this type of system too. Learning the basics of data-driven systems includes concepts such as data values, attributes, data types, data collection, data cleaning, data processing, data storage, data visualization, and using data. It could be considered particularly relevant that primary aged children start to understand the potential bias and trustworthiness of Artificial Intelligence (AI) and machine-learning systems (European Commission 2019). Trustworthiness in AI and data includes human agency, safety, privacy, transparency, diversity, non-discrimination and fairness, societal and environmental wellbeing and accountability. Teaching about data, particularly AI and machine learning, to primary aged children has not been well researched; therefore, classroom vocabulary, concepts, progression, pedagogy, and assessment will require careful design.

C5 - Programming Languages

Fundamental to digital technologies is the concept of programming languages. Programming languages are used to implement designs or solutions which solve problems. Children should develop a high degree of familiarity with and a deepening understanding of the three fundamental programming constructs by learning about and using programming languages. These constructs are sequence, repetition, and selection. An essential aspect of learning about programming languages is the difference between the syntax of the language and the semantics. Like learning about spoken languages, there are a set of pre-defined words, commands, which must be spelt correctly and have grammatical rules about how to assemble them. Simply put, these rules are the syntax of a programming language. Conversely, just as in a spoken language, words, sentences, paragraphs, and texts have a meaning, so commands, groups of commands and programs have meaning.

C6 - Problem – solving Approaches

When solving problems in computing contexts, a popular coined phrase is Computational Thinking (CT). The term, CT, is often attributed to Seymour Papert (Papert 1980), who mentioned the term when describing children's thinking as they used programming to solve problems. There are several common concepts and approaches which might be expected to be seen in a primary digital technologies' curriculum at any point in time (Millwood *et al.* 2018). Therefore, at this point in time, there may be some pressure to call concept C6' Computational Thinking', rather than 'Problem-solving approaches'. However, it is the authors view that the popularity of the term CT may wane and change in line with research as the topic unfolds. For example, problem solving in Machine Learning activities is very different to in traditional programming. Consequently, we have included programming constructs and concepts, sometimes classified as 'Computational Thinking', in C5 'Programming languages'. In 'Problem-solving approaches', concepts such as abstraction, decomposition, generalisation (or spotting patterns) and logical reasoning should be included. Problem solving involves using strategies such as breaking down problems into parts, interpreting patterns and models, and designing and implementing algorithms. When developing classroom problem solving activities, it is important to ensure the concepts are linked to digital technology contexts so that children know how to apply problem solving in practice.

Processes

There are eight themes in the *Processes Dimension* as shown in Figure 3.

Figure 3: Processes Dimension – Process and Productions Skills

P1 Using

P2 Managing Requirements and Defining Problems

P3 Designing and Implementing

P4 Developing and Using Representations

P5 Debugging and Testing

P6 Reasoning and Evaluating

P7 Communicating, Presenting and Sharing

P8 Collaborating and Teamwork

P1 – Using

Children should use digital technologies to achieve purposeful outcomes in various relevant contexts. "Using" can manifest in many ways, for example, when children are investigating, following or exploring. Using applies across all the primary digital technologies concepts. As digital technologies are constantly evolving and changing, children should be given early opportunities to use new and emerging types of technologies and systems, i.e., mobile devices, desktop computers, robotics and other physical computing devices. Machine learning and AI also provide opportunities for children to engage with technologies that directly impact their lives. Children should investigate how people safely and ethically use common digital systems to engage with information, communication, and recreation needs. A child exploring as a "user" should also highlight their experience, perceptions, and expectations of technology to help them think more carefully about its design and implementation for others. For example, children should consider how data is entered, or information is presented so that it is accessible for all people.

P2 - Managing Requirements and Defining Problems

Defining requirements and identifying problems includes thinking about the people (the users), what systems (hardware and software) are needed, what data will be used or created, the requirements (the goal), and any possible constraints. Managing requirements and defining problems are closely related skills and, in the primary school context, may also be called tasks, activities, expressing ideas, plans and even exploring. Defining problems includes stating an area of interest (sometimes called the domain or scope), who is involved (the users), identifying possible constraints, and the possible solution required. Requirements can be called other things, such as goals, task definitions, problem outlines and aims. The recognition, definition, evaluation, management and communication of requirements apply to any digital technology problem and can be completed independently or with others. For example, there are requirements to be considered when making a digital presentation, developing a computer game, finding things out by searching online for information, or gathering data to answer a question. Classroom activities may also include using requirements developed by others, then adding changing and adapting the requirement to create a new requirement to solve a different problem.

P3 - Designing and Implementing

Designing and implementing could have been divided into two separate processes and production skills. However, the line between these two can be blurred, depending on children's approach to developing products. Designing requires children to develop and use representations (see P4) and naturally depends on managing requirements and defining problems (see P2). Designing and implementing applies across all primary digital technology concepts. It can be done as one phase for an entire project, or a task can be decomposed into parts, and each part can be designed and implemented in turn. Children should create designs that include thinking about; algorithms, data, and systems, while making sure their designs meet the needs of all users.

P4 - Developing and Using Representations

Developing and using representations underpins all process and production skills and the development of understanding of any concept. Working with representations is sometimes called abstraction. When abstracting, we ignore detail that we do not need and only include the specific details to complete the task. An algorithm is an abstraction; it represents the precise steps or rules needed to solve a task. For example, we might have a set of verbal instructions of an unplugged activity (Bell and Vahrenhol 2018) of how to walk a square shape. The representation of the algorithm in an unplugged activity may be: "take one step, turn a right angle to your left, take another same size step, turn a right angle to your left, repeat these two more times". Our representation has not included much detail about how far each "same size step" is nor where we started to take our first step. This detail was not crucial for this representation. As we implement this representation of our algorithm as a program, for instance in a block-based programming language such as 'Scratch', we need to think about the specific detail that was not decided upon in our unplugged verbal representation.

P5 - Debugging and Testing

Debugging and testing could also be included in the process and production skill of 'Reasoning and Evaluating' (see P6). However, developing this skill is crucial in digital technologies; therefore, merits a distinct process. Simply put, a bug is when we notice something is not quite right with an algorithm or program. The algorithm or program does not do what we expect.

However, this simplistic view hides many reasons that might have caused the bug. Reasons include not understanding how the programming language commands work, using the wrong command to implement a step in the design, a problem with the algorithm, or another design aspect. To debug an algorithm or program, we have to find and fix the bug. Children should be encouraged to expect bugs and be patient and resilient in finding and fixing them. Debugging describes a reaction to something wrong, whereas testing is planned. Testing is designing specific tests to ensure that the solution will meet the requirements. Testing is the skill to anticipate and find potential bugs before they happen. Children might ask their friends to "break" their final solution during testing, encouraging them to think of all the unexpected things that could happen. As children become more familiar with the different types of bugs they encounter, they can start to be more systematic about testing.

P6 - Reasoning and Evaluating

Reasoning is how children use existing knowledge to predict an outcome and then analyse what occurs to change or confirm their ideas. By reasoning, children can justify explanations and interpretations. On the other hand, evaluation requires children to compare something against a set of criteria. After making a comparison, children are likely to use reasoning to make judgements about the aspect that is being evaluated. In teaching primary-level programming, care should be taken in teaching children to evaluate one particular concept, the efficiency of programs. However, it is unlikely that primary-aged children will be able to evaluate the efficiency of the programs. It is more appropriate that children learn to evaluate the readability, reuse, and elegance of the programs they encounter.

P7 - Communicating, Presenting and Sharing

Communicating, presenting, and sharing are general capabilities that support problem-solving skills needed by children when they work with digital technology. Children will present their ideas and information and learn about different forms of representations to suit different needs (see P4). In doing this, children will learn about and start to act ethically, safely, and responsibly. When learning about ethics, children will reflect on digital technologies with regard to diversity, discrimination, and fairness. With respect to safety, children will start to think about potential environmental, societal, and personal harm that systems can cause. About responsibly, children will start to consider where data has originated from, who has developed systems and be introduced to the notion of ownership and accountability.

P8 – Collaborating and Teamwork

Sometimes included as a Computational Thinking approach (Millwood *et al*. 2018), collaborating and teamwork are process and production skills which should be fostered not only to support general problem solving but also to gain specific digital technologies skills. For example, pair programming is a standard industry technique and working in teams to complete projects is common. However further research is needed on the best way to support young children to use collaboration effectively in digital technology activities.

5. Considerations for Teaching and Learning

This document does not provide a review of pedagogy for teaching digital technologies in primary classrooms, but when mapping statements to themes, certain approaches became evident and could be easily incorporated.

Three broad approaches to learning were noted, exploration, problem-solving and making, and direct instruction. Learning about digital technology is a complex subject with some technical vocabulary. Terminology in digital technologies can mean different things depending on the school subject being learned. For example, a variable in science or maths means something different in programming (Herman *et al.* 2018). Therefore, a general trend in research and practice is towards a blended pedagogy that includes some direct teaching to introduce concepts and vocabulary along with exploration, problem-solving and making to put concepts into practice. Linking the abstract to concrete and vice-versa is important for children to form accurate mental models and avoid misconceptions that will be hard to shift as children learn about topics in depth.

Approaches such as pair programming, peer instruction, and teacher modelling encourage collaboration and discourse. Learning together may be particularly important for very young children as classroom talk helps children build knowledge, and discourse reveals learner thinking for formative assessment. Classroom activities that require children to read code before they write code have been found to be essential for learning to program. Similarly, sequences of specific approaches are very effective in teaching programming, such as 'Use-Modify-Create' and 'PRIMM' - Predict, Run, Investigate, Modify, Make (Sentance *et al*. 2019). These approaches help children learn what is possible as they use a project created by someone else and then take gradual ownership as they use it, read it, answer questions about it, modify it and then make something new for themselves.

6. Applying the PC Model in Curriculum Development

The next steps for using the proposed *CP Model* are to elaborate the Concept and Process themes as more detailed statements. These statements will inform the development of learning outcomes for primary-aged children.

As guidance for curriculum designers, each theme of the *CP Model* does not hold the same weight. Children at different points in progression may spend more or less time on different themes, for example, younger children may spend more time on *P1 Using* and *C2 Systems* as they are finding out what is possible when using software. However, curriculum designers may decide that it is important for children to quickly start making their own artefacts and balance *P1 Using* with *P3 Designing and Implementing*. Other approaches may include introducing activities which promote *P6 Reasoning and Evaluating*, for example, asking children to predict what programs do, and introducing code comprehension early. This will depend on the pedagogy that is being adopted. It is important to ensure that the knowledge and understanding of concepts are included as learning outcomes as well as the process and productions skills that put them into practice otherwise there is a risk that practical activities are not linked to the underlying concepts that are being introduced.

The *CP Model* serves only as broad headings to support elaboration rather than acting as a constraint. For example, there is no requirement to say that for every pupil in any year that they must encounter a learning outcome for each and every combination of the *CP Model*. Children should follow a path of learning outcomes as a progression for each theme. What this progression looks like will depend on many things, including theories related to how children learn, including what misconceptions are likely and must be overcome, pedagogy, and teacher knowledge and confidence.

7. Concluding remarks

In producing the *CP Model*, we have developed a method for grouping theoretical competency model statements and curricula statements from school pilots. This method involved the grouping of statements into themes which after multiple iterations, outlined in the accompanying technical report, became the dimensions of the *CP Model*. As well as mapping these dimensions we have also highlighted the importance of contexts and have suggested opportunities to link to the *Draft Primary Curriculum Framework* key competencies.

The final suggested model includes six concept themes and eight processes and production themes as shown in Figure 1. The *CP Model* can be used in next steps for developing learning outcomes and progression maps. As a new area of learning for primary children, digital technologies have been implemented in quite different ways across the world, how effective each of these approaches has been, is still to be evidenced.

As expert practitioners, policymakers and researchers work together to develop the curriculum different views will be heard which will need to be carefully evaluated. International research should take a high profile in supporting this evaluation. Although practically applied research in primary computing is limited, many new ideas are currently being developed to improve the teaching and learning of primary digital technologies, these should be taken advantage of in the creation of the Irish iteration of digital technology in the Primary School Curriculum.

Reference List

Bell T. and Vahrenhold J. (2018) *CS Unplugged—How Is It Used, and Does It Work?*. In: Böckenhauer H,J., Komm D., Unger W. (eds) *Adventures Between Lower Bounds and Higher Altitudes*. Lecture Notes in Computer Science, vol 11011, Switzerland: Springer Cham, available: https://link.springer.com/chapter/10.1007%2F978-3-319-98355-4_29 [accessed 12 May 2021].

Computing Crossroads (2021) *Highlighting Career Diversity in Computing*, available: https://computingcrossroads.org/ [accessed 10 September 2021].

European Commission (2019) *Ethics Guidelines for Trustworthy AI - High-Level Expert Group on Artificial Intelligence*, *EU*, available: https://www.aepd.es/sites/default/files/2019-12/ai-ethics-guidelines.pdf [accessed 08 January 2021].

European Commission (2020) *DEAP - Digital Education Action Plan 2021-2027*, EU, available: https://ec.europa.eu/education/sites/default/files/document-library-docs/deap-communication-sept2020_en.pdf [accessed 25 September 2021].

Hermans, F., Swidan, A., Aivaloglou, E. and Smit, M. (2018) Thinking out of the box: comparing metaphors for variables in programming education. *Proceedings of the 13th Workshop on Primary and Secondary Computing Education* (online), available from: https://doi.org/10.1145/3265757.3265765 [accessed 25 May 2021].

Massachusetts Institute of Technology (2015) *Logo programming*, Boston: MIT, available: https://el.media.mit.edu/logo-foundation/what_is_logo/logo_programming.html [accessed 18 January 2021].

Massachusetts Institute of Technology (2021) *Scratch programming*, Boston: MIT, available: https://scratch.mit.edu/ [accessed 18 January 2021].

Millwood, R., Bresnihan, N., Walsh, D. and Hooper, J. (2018) *Review of Literature on Computational Thinking*, Dublin: NCCA, available: https://ncca.ie/en/resources/primary-coding_review-of-literature-on-computational-thinking/ [accessed 21 May 2021].

National Council for Curriculum and Assessment (2020) *Draft Primary Curriculum Framework*, Dublin: NCCA, available: https://ncca.ie/en/resources/ncca-primary-curriculum-framework-2020pdf/ [accessed 15 June 2021].

Papert, S, A. (1980) *Mindstorms: Children, computers, and powerful ideas*, New York: Basic books Inc publishers, available: https://mindstorms.media.mit.edu/ [accessed 18 January 2020].

Sentance, S Waite, J and Kallia, M (2019) *Teaching computer programming with PRIMM: a sociocultural perspective*. Computer Science Education (online), 29(2), pp.136-176, London: available from: https://www.tandfonline.com/doi/abs/10.1080/08993408.2019.1608781 [accessed 7 May 2021].

Tedre, M., ed. (2014) *The Science of Computing*, Florida: CRC Press, available: https://www.perlego.com/book/1509964/the-science-of-computing-pdf [accessed 20 July 2020].

Jane Waite
Keith Quille