Original Research Article

Journal of Educational Computing Research
2023, Vol. 0(0) 1–24

DOI: 10.1177/07356331221133822
journals.sagepub.com/home/jec
SAGE

What Else Can Be Learned When Coding? A Configurative Literature Review of Learning Opportunities Through Computational Thinking

Ana Melro[1,2], Georgie Tarling[3], Taro Fujita[3], and Judith Kleine Staarman[3]

Abstract

Underpinning the teaching of coding with Computational Thinking has proved relevant for diverse learners, particularly given the increasing demand in upskilling for today's labour market. While literature on computing education is vast, it remains unexplored how existing CT conceptualisations relate to the learning opportunities needed for a meaningful application of coding in non-Computer Scientists' lives and careers. In order to identify and organise the learning opportunities in the literature about CT, we conducted a configurative literature review of studies published on Web of Science, between 2006 and 2021. Our sample gathers 34 papers and was analysed on NVivo to find key themes. We were able to organise framings of CT and related learning opportunities into three dimensions: functional, collaborative, and critical and creative. These dimensions make visible learning opportunities that range from individual cognitive development to interdisciplinary working with others, and to active participation in a technologically evolving society. By comparing and contrasting

[1]Technology and Intermedia Studies Research Centre, University of Maia, Maia, Portugal
[2]Communication and Society Research Centre, University of Minho, Braga, Portugal
[3]School of Education, University of Exeter, Exeter, UK

Corresponding Author:
Ana Melro, Instituto Universitário da Maia, Castelo da Maia, Maia 4475-690, Portugal.
Email: amelro@umaia.pt

frameworks, we identify and explain different perspectives on skills. Furthermore, the three-dimensional model can guide pedagogical design and practice in coding courses.

Keywords

learning opportunities, computational thinking, coding, non-CS education, literature review

Introduction

The idea that more people should learn coding, programming or computing, as they are interchangeably used (Bocconi et al., 2016), has become a global mainstream discourse. In particular, higher education students in non-Computer Science (CS) fields, from social sciences to humanities, are being encouraged to learn coding so they can play a more effective role in solving today's challenges (Burke et al., 2016; Grover & Pea, 2017; Popat & Starkey, 2019). Not only have many of these students missed being taught computing as part of their compulsory education (Lockwood & Mooney, 2018), but they are also about to enter a demanding job market in terms of "the skills they need to compete successfully in the global digital economy" (Davenport et al., 2019, p. 3). Coding is promoted as key to economic success due to the growth in emerging tech jobs and because it is essential for all jobs (Orlik, 2019). Alongside the employability rationale, it is also argued that a basic level of computing understanding is becoming ubiquitous in terms of interdisciplinary conversations (Chilana et al., 2015; Dawson et al., 2018). Traditional disciplines are being transformed by computation and new ones are emerging, such as digital humanities or computational social science (diSessa, 2018). Furthermore, there is a recognition of the need to support individuals to be not just problem solvers and programmers, but also thinkers and active citizens for a more inclusive society and democracy (Boyd & Crawford, 2011; Grover & Pea, 2017). From this perspective, learning code can also be a way to critique its usage (Dufva & Dufva, 2016).

A prevailing argument when introducing coding to diverse learners is, that this should not be limited to teaching a programming language, but should also include the development of wider skills (diSessa, 2018; Dufva & Dufva, 2016; Grover & Pea, 2017; Turner, 2019). As Grover and Pea (2017, p. 21) state, "teaching mathematics has moved towards thinking like a mathematician", therefore teaching computing should also privilege Computational Thinking (CT). Although CT is a thought process independent of technology (Bocconi et al., 2016), and does not therefore need to be taught using a computer, research shows that introducing CT and teaching across disciplines, can make programming more appealing and accessible for non-CS learners (Dawson et al., 2018; Yadav et al., 2011). The idea of CT has its roots in Seymour Papert's (1980) constructionist theory, in which programming was seen as a way of developing children's problem-solving and analytical skills by making the abstract concrete. Over 20 years later, the notion of CT comprising higher order thinking skills transferable to other fields became widely disseminated when Jeannette Wing (2006) argued that CT develops essential skills "for everyone, everywhere." Since then, CT has been explored extensively in the literature through a variety of approaches, conceptualisations and emerging terminologies. However, there remains a lack of consensus around definitions of CT (Grover & Pea, 2013; Israel et al., 2015; National Research Council, 2011), particularly about the learning opportunities it opens up for students from non-CS backgrounds who are introduced to coding. As multiple frameworks are being adopted, skills are being framed differently, leading to confusion about what students are learning and how this is being assessed. This paper therefore seeks to make visible and discuss the diversity of learning opportunities for students learning coding that are associated with different conceptualisations of CT across the literature by addressing the following research question: (RQ) *What types of learning opportunities are revealed by systematically mapping the CT literature?*

In this study, we have undertaken a configurative review of the literature and mapped the learning opportunities related to CT to create an organisational model. This model consists of three dimensions: (1) a *functional* dimension in which framings of CT focus on operational skills related to problem-solving (eg. decomposition, debugging); (2) a *collaborative* dimension which emphasises learning to participate and work with others in communities and interdisciplinary teams (eg. expressing, communication); and (3) a *critical and creative* dimension which focuses on learning to imagine possibilities and question the impact of coding in order to make informed and inclusive decisions in society (e.g., evaluation, empowerment).

The intention here is not to propose a new theoretical CT framework itself, but rather to identify and explain the crossovers and nuances between different conceptualisations and related learning opportunities. The main contribution of this paper therefore is to make visible the diversity of learning opportunities identified across the range of CT framings. The proposed model can help teachers and designers of coding courses in their pedagogical design, and will be of particular value for those seeking to meet the rising demand from students in non-CS fields (Camp et al., 2017; Dawson et al., 2018; Sax et al., 2017).

Configurative Review of the CT Literature

We conducted a systematic configurative review (Gough et al., 2017) to identify key themes or dimensions across the CT literature. For the sample selection, we searched for the term "Computational Thinking"[1] on April 25th of 2021, using Web of Science (WoS - http://www.webofknowledge.com) from Clarivate Analytics, since it provides access to multiple academic databases across disciplines. The search included articles only in English language and within the timeframe 2006–2021, since the expression "Computational Thinking" was first coined by Jeannette Wing in 2006. The first hit resulted in a total of 1984 articles. The sample selection then followed four criteria as described in Table 1, gathering a total of 34 articles (N = 34).

Table 1. Criteria Followed in the Sample Selection.

Criteria	Description	Number of articles
First hit	Search by keywords "computational thinking" from 2006 to 2021	1984
#1	Selection by types of documents	1822
#2	Selection by research areas and WoS categories	985
#3	Selection by words in the title	314
#4	Manual selection by relevance (reading of the abstract)	34

The first criterion consisted in filtering by types of documents. Since conference proceedings are relevant in this field, we chose to include them. Literature reviews are also significant to understand what has already been considered when conceptualising CT. Thus, research articles, proceedings papers and literature reviews were included while types of documents excluded encompass book reviews, letters, corrections, bibliographic items, book chapters and reprints.

For the second criterion we filtered the search by research areas and WoS categories. We included the following research areas: "Education Educational Research," "Computer Science" and "Social Sciences Other Topics." As for the WoS categories, we included articles within "Education Educational Research," "Computer Science Theory Methods", "Computer Science Interdisciplinary Applications" and "Social Sciences Interdisciplinary." All remaining areas and WoS categories were excluded to ensure that articles collected came from an educational or an interdisciplinary research field. Despite filtering by subject, there were still articles crossing over with specialised areas, such as robotics or mathematics, leading to another round of filtering.

Since the database exported from WoS contained empty fields in the columns referring to "keywords" and "abstract," we opted to refine the search by including and excluding specific words from the title of the article. By doing so, we ensured that the selected articles were closer to the aim of the current study. Therefore, we included in the title the words "computation" or "computational," and excluded for instance any term related to robotics, mathematics, modelling, games, physics, biology and specific software or programming languages. After the third criterion, our sample was reduced to a total of $N = 314$ articles.

The final criterion involved three rounds of manual peer revision. This consisted firstly of three team members independently reading each of the 314 articles' abstract and deciding to include or exclude them, according to their relevance to our study. Secondly, the team discussed and compared each member's selection. There were no absolute disagreements about each selection but there were occasionally some differences of opinion. In these instances, it was agreed that each reviewer would revisit the full article. Thirdly, having all read the articles, joint decisions were made to agree on the final sample. Our priority was to include papers that would discuss definitions of CT and the range of related learning opportunities about the teaching of coding, regardless of the research methods employed in each paper. We therefore did not exclude

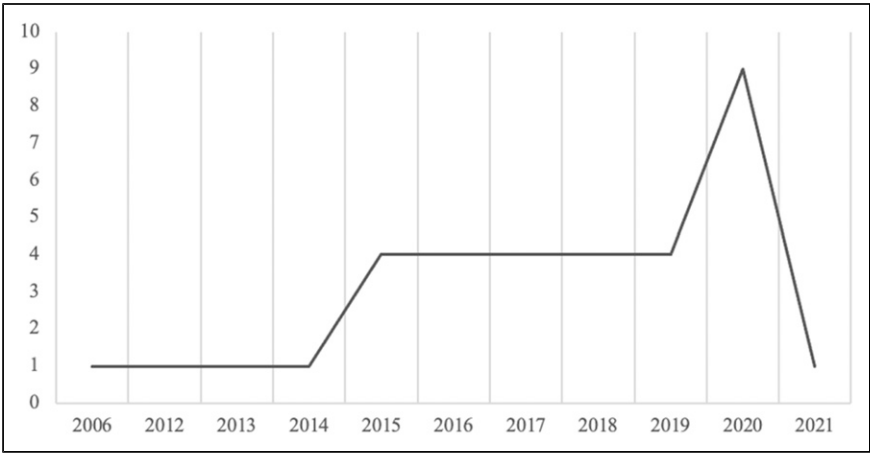

Figure 1. Number of papers in the sample from 2006 to 2021 (N = 34).

any paper based on its research method. In this study, we draw from the labels ‘skills,’ ‘competencies,’ ‘components’ or ‘facets’ to describe the learning opportunities addressed by the authors. We included 34 articles (Appendix 1) consisting of: literature reviews on CT (mostly on CT definitions and related skills); cross-field definitions or new conceptualisations of CT; and conceptual and empirical studies about CT skills. We then excluded studies on pedagogic practices or CT applications, articles about STEM education, and empirical studies (practical implementations, assessments, evaluations, etc.). As a final stage, we looked at the references in our sample papers to ensure we did not miss important frameworks. The sample was then analysed using NVivo.

Sample Characterisation

As shown in Figure 1, literature about CT conceptualisations appears to have an exponential growth on the number of papers at least from 2014 to 2020. In our sample (N = 34), 47% of the papers are studies about primary and secondary education, 29% are about secondary education and 26% discuss CT in higher education.

Studies derived mostly from North America, in particular from the US (n = 16; 47%), as observed in Figure 2. The second popular country is Turkey (n = 4) in Middle East, and only one study was found from the UK within European perspectives (n = 6).

The papers in our sample allowed us to identify patterns related to CT conceptualisations and related learning opportunities which were then organised into three main key dimensions. A summary of the configurative literature review is displayed in Appendix 1, where we describe in detail the main contributions of each study, their conceptualisations of CT and related learning opportunities.

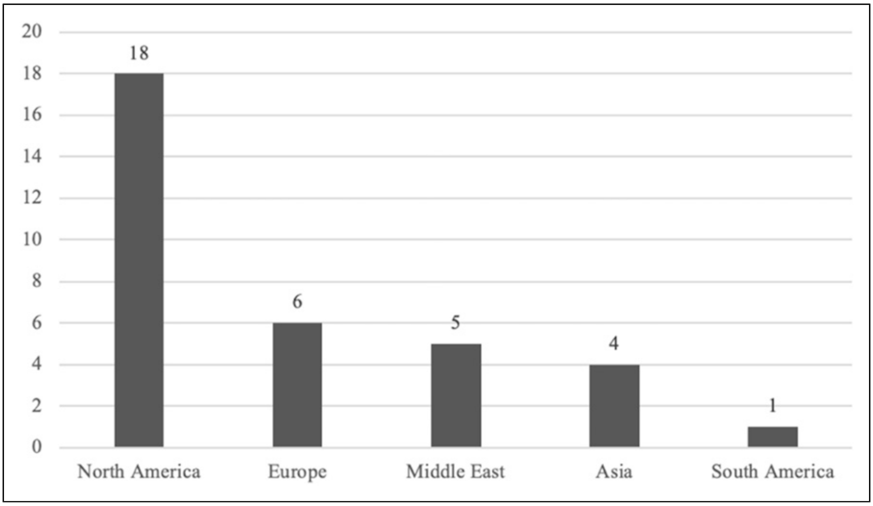

Figure 2. Frequency of papers across the world (N = 34).

Key Dimensions in the CT Literature

To get an overview of the frameworks, terminologies, and crossovers around CT conceptualisations we classified the papers in four main types of study (Appendix 1):

A – Creation of a new or alternative CT framework;
B – Comparison between CT and other frameworks;
C – Inclusion of CT in the context of a broader conceptualisation;
D – Overview of definitions of CT identifying similarities and differences.

In type A, the authors propose an alternative CT framework, sometimes using new terminologies, such as *computational participation* (Kafai, 2016), *computational empowerment* (Iversen et al., 2018), or *critical computational literacy* (Lee & Soep, 2018). In type B, authors compare or combine CT with other frameworks or disciplines, such as Digital Literacy, Media and Information Literacy (MIL), or Problem Based Learning (PBL). In this type, authors commonly built on other frameworks, being among the popular ones, the Brennan and Resnick's (2012) framework, which is composed of *computational concepts*, *practices* and *perspectives*. The framework of the International Society for Technology in Education (ISTE) and the curriculum of the Advanced Placement Computer Science Principles (CSP) were also central to some of the papers analysed. These two frameworks were included in our analysis because they were frequently cited in our sample papers. Therefore, we refer to them as important landmarks in the conceptualisation of CT. Thirdly, type C refers to papers in which CT is part of a broader conceptualisation such as the Danish framework of "Technological

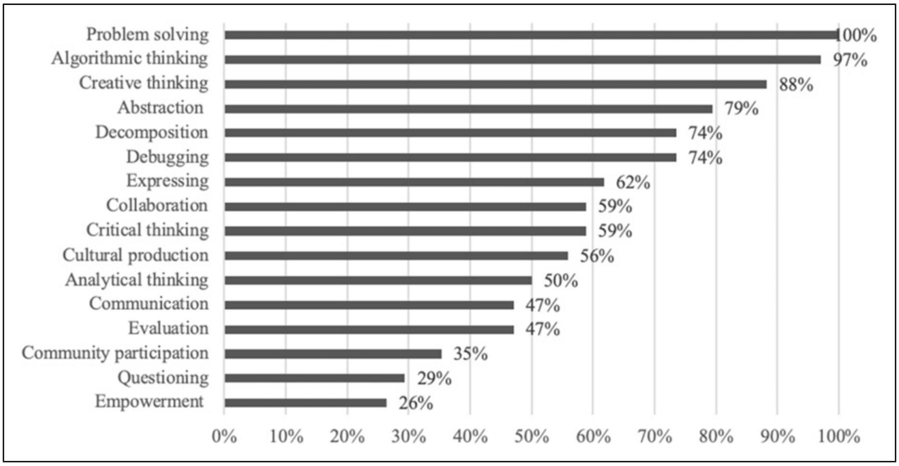

Figure 3. Top 16 most frequent learning opportunities identified in the sample (N = 34).

Understanding" (Caeli & Bundsgaard, 2020). Lastly, in type D, authors present an overview of key competences and definitions associated with CT, usually using systematic reviews of the literature. For example, Shute et al. (2017) reviewed 45 papers containing a diversity of definitions and models of CT in which they compile a range of CT 'core facets'. Cutumisu et al. (2019) scoped 39 papers on CT assessment and identified the most commonly assessed skills or computational 'concepts.' Likewise, Hasesk and Ilic (2019) systematically reviewed the literature to identify the most assessed CT skills, which is useful in our study to get an overview of existing learning opportunities.

Finally, in order to identify the key main themes or dimensions that resulted in our three-dimensional organisational model, we looked at the learning opportunities revealed in each paper (based on labels such as 'skills', 'competences', 'components', 'facets') and went through a three-stage process. First, we scraped a list of the most frequent words related to skills from our sample on NVivo. Second, we refined this to a list of 16 discrete learning opportunities based on the skills mentioned in each paper. As observed in Figure 3, "problem solving" appears across all papers in our sample (N = 34), followed by "algorithmic thinking" and "creative thinking". At the bottom of the list, the least mention learning opportunities are "empowerment," "questioning", and "community participation."

We then conducted a thematic analysis arriving at three overarching dimensions that best described the range of learning opportunities found in the different framings of CT: *functional*, *collaborative*, and *critical and creative*. The label *functional* was chosen to encompass ways of approaching CT that focus on fostering operational or 'required' skills so one can 'function'[2] in a changing society and future workforce (Dufva & Dufva, 2016). This focus describes specific cognitive processes involved in problem

Table 2. CT and the Functional Dimension.

Dimension	Summary	Conceptualisations
Functional	Framings of CT which draw from the classic or original conceptualisation and focus on cognitive skills (core skills) related to problem-solving.	Cognitive CT (Kafai et al., 2019) Core CT facets (Shute et al., 2017) Core competencies (Voogt et al., 2015) CT concepts (Brennan & Resnick, 2012) Cognitive processes (Wing, 2006) Core CT concepts (CSTA/ISTE) (Barr & Stephenson, 2011) CT elements (Grover & Pea, 2013)

formulation and solving (Cuny et al., 2010; Shute et al., 2017) and is echoed in many institutional discourses with an economic-driven rationale for the teaching of coding (Williamson, 2016). We chose the label collaborative as it acknowledges coding as a social practice at different scales: both at a micro level of the classroom and macro level of community of practice (diSessa, 2001; Dufva & Dufva, 2016; Iversen et al., 2018; Turner, 2019). The label critical and creative was informed by MIL frameworks which emphasise the dynamic between critical and creative thinking skills in supporting citizens to engage in a digital world (Gretter & Yadav, 2016; Iversen et al., 2018; Kafai et al., 2019; Knochel & Patton, 2015; Lee & Soep, 2018). The following headings describe in detail the learning opportunities we associate with each dimension based on different framings of CT found in the literature. At the end of each section, we summarise the main CT conceptualisations and the learning opportunities drawn from the work of other scholars.

Findings

Functional Dimension

The first dimension in our mapping is defined as functional. The conceptualisations organised into this dimension describe foundational elements of CT which emerged across our sample (Table 2).

In this dimension, CT is related to operational problem-solving with computers, using mathematical and engineering thinking as a foundation (Shute et al., 2017). Studies about CT, including all papers in our sample, start with the conceptual understanding first described by Wing (2006), which involves five main cognitive processes: problem reformulation, recursion, problem decomposition, abstraction and systematic testing. This makes it the dominant framing of CT focussing on key learning opportunities regarding individual cognitive processes for problem-solving (Kafai et al., 2019). However, there remains debate across our sample around what the 'core skills' are. In a review of the CT literature, Shute et al. (2017) summarise these as the 'core CT facets': decomposition, abstraction, algorithms and debugging, iteration and generalisation. Through their model, the authors emphasise the importance of approaching problems in a systematic way: by breaking down a complex problem into smaller parts (problem decomposition); through iterative debugging; by finding patterns and generalise solutions (abstraction/pattern generalisation); and through algorithm design for solving problems (Shute et al., 2017).

Whilst there is much crossover between the key or core skills identified, there are also nuances regarding the specific cognitive skills identified as being stages in this process and in the terms used to describe a similar skill. In addition, the number of these core components fluctuates depending on the framework. Nonetheless, these cognitive processes translate into valuable learning opportunities that usually comprise analytical reasoning for problem decomposition, algorithmic thinking, abstraction, and automation (Yadav et al., 2016). One mainstream CT framework acknowledged by several authors in our sample is the Computer Science Teachers Association (CSTA) and International Society for Technology in Education (ISTE), in which Barr and Stephenson (2011) list nine 'core CT concepts': data collection, data analysis, data representation, problem decomposition, abstraction, algorithms and procedures, automation, parallelisation, and simulation. Drawing from the ISTE and Wing's definitions, Grover and Pea (2013) also summarise nine widely accepted CT elements for secondary education: abstractions and pattern generalisations; systematic processing of information; symbol systems and representations; algorithmic notions of flow of control; structured problem decomposition; iterative, recursive, and parallel thinking; conditional logic; efficiency and performance constraints; and debugging and systematic error detection.

Another popular model revealed in our sample is the Brennan and Resnick's (2012) CT framework, in which CT core components are described as computational 'concepts', i.e., the concepts designers engage with as they program, such as iteration, parallelism, etc. For instance, in a scoping review, Cutumisu et al. (2019) found that most studies focus on interventions that promote CT concepts and practices that mainly assess algorithmic thinking, abstraction, problem decomposition, and logical thinking. Also in Hasesk and Ilic's (2019) literature review, abstraction, algorithmic thinking, decomposition and sequence of steps were the most assessed CT skills.

Overall, in the functional dimension, CT is often described in terms of problem formulation and solution through cognitive processes (Taslibeyaz et al., 2020). In some articles in our sample, however, CT is not just related to cognitive skills, but focusses also in more general thinking and in the "habits and dispositions needed to solve complex problems" (Voogt et al., 2015, p. 720). Using Bloom's taxonomy, Tang et al. (2020a), for example, conducted a literature review to classify the most common keywords associated with CT in terms of cognitive, affective and psychomotor domains. In the cognitive domain, the authors consider some of the core skills, such as problem-solving, abstraction and decomposition, but also include a mix of general thinking skills such as reasoning, reflection, metacognition, and evaluation (Tang et al., 2020a). Attitudes and affective dispositions are used in other framings of CT as

Table 3. CT and the Collaborative Dimension.

Dimension	Summary	Conceptualisations
Collaborative	Framings of CT that highlight the value of collaborative work and the skills needed to engage in teams and participate in coding communities. It also addresses interdisciplinarity and dialogue across disciplines in discussing solutions to problems.	Computational participation (Kafai, 2016) Computational communities (Lachney, 2017) Situated CT (Kafai et al., 2019) Computational practices and perspectives (expressing and connecting) (Brennan & Resnick, 2012) CT across subjects (Barr & Stephenson, 2011)

complements to the cognitive domain of CT (Sondakh et al., 2020; Tang et al., 2020a, 2020b). Although attitudes can impact students' engagement with learning, it is difficult to distinguish specific learning opportunities offered in the context of CT (Velázquez-Iturbide, 2018).

To some extent, the skills revealed in the functional dimension describe the granular level of deconstructing the problem-solving process which translates into vital cognitive learning opportunities for students who are introduced to coding.

Collaborative Dimension

The second dimension in our mapping is defined as collaborative. The conceptualisations organised into this dimension (Table 3) describe elements of CT that acknowledge the importance of social interactions when coding with others and therefore builds on interpersonal and communication skills. It focuses on the need to prepare learners for interdisciplinary and inclusive collaborative workplaces, where teamwork and communication are increasingly seen as important skills to actively participate in the community.

The collaborative dimension encompasses what Kafai (2016) conceptualised as *computational participation* and then incorporated in the *situated* framing of CT (Kafai et al., 2019). This draws from constructionist learning theories and focuses on building meaningful social relationships through coding as an alternative proposition to the *cognitive* framing (Kafai et al., 2019). Coding is therefore seen as a collective process that is developed through peer-supported activities and sharing digital artifacts in online or offline collaborative spaces, such as hackathons, coding clubs or GitHub. For the authors, whereas the *cognitive* framing of CT uses an algorithmic lens towards problem-solving, *computational participation* extends the thinking beyond the individual to integrate social networks and digital tools in a networked society (Kafai, 2016; Kafai et al., 2019). Using collaborative online spaces, leveraging existing resources or remixing others' work are also what Brennan and Resnick (2012) refer to as computational practices. But in order to take part in these practices, there needs to be an awareness of the skills needed to engage in collaborative work.

Across the literature in our sample various authors include skills related to learning coding in a team as part of their definition of CT. For example, the ISTE framework states that in addition to being able to break problems down together into smaller parts (*decomposition*) and to simplify concrete into general solutions (*abstraction*), learners should be able to negotiate within a team the merging of parts of the solution into the whole (*negotiation*), as well as, building group solidarity behind one idea or solution (*consensus building*) (Barr & Stephenson, 2011). Drawing on the ISTE framework, Korkmaz et al. (2017) include cooperative learning, which they define as working in small groups towards a common goal, as one of five important dimensions in their CT assessment scale. They also highlight communication as a foundational competence, not only for cooperative learning, but also for problem-solving and critical thinking (Korkmaz et al., 2017).

In Brennan and Resnick's *computational perspectives*, computation is seen as medium of communication or as means of *expressing* (pointing, clicking, browsing, and chatting), sharing things or *connecting* with others in the digital world. Developing relationships with others is articulated in Brennan and Resnick's CT framework as a way to learn "the value of creating *with* others, and the value of creating *for* others" (2012, p. 10), fostering both inclusive collaborative work and community engagement through the creation of digital artefacts. Using Brennan and Resnick's *perspectives* in the conceptualisation of *programming empowerment*, Kong et al. (2018) found that primary school students with positive attitudes towards collaboration had greater creative self-efficacy. Also building on *computational perspectives,* Pinkard et al. (2020) explored how students shifted their perceptions of CT through knowledge and experience, such as project work. The authors found that teamwork and building community (*computational participation*) offered students new computational perceptions about themselves (*computational identity*) and the domain.

The conceptualisations of CT mapped to the collaborative dimension are also informed by an understanding that an interdisciplinary and inclusive approach is needed for solving some of the world's wicked problems from social inequities to climate breakdown. As Kafai et al. (2019, p. 104) argue, "fostering personal connections alone is no guarantee for inclusion." From this perspective having a basic understanding of coding and its vocabulary is important not only for those wanting to programme computers themselves, but also as a communication skill for those wanting to engage in conversations with programmers, work in interdisciplinary teams or learn from other how computational thinking might be applied to their own studies or problems. These definitions recognise the need for CT to be in dialogue with contextual cultural and subject-based knowledge. For Lachney (2017), allowing students to explore aspects of their culture with coding provides not only opportunities for understanding CT vocabulary and concepts, but also for reflecting on their own cultural capital and heritage. A key aspect of CT in this context is being able to create diverse entry points for others

Table 4. CT and the Critical and Creative Dimension.

Dimension	Summary	Conceptualisations
Critical and creative	Framings of CT that focus on the importance of developing critical and creative thinking while engaging with coding or in conversations about coding. This includes materialising creative ideas to make informed decisions concerning ethics, social justice and the impact of coding in society.	Computational empowerment (Iversen et al., 2018) Critical computational literacy (Lee & Soep, 2018) Critical CT (Kafai et al., 2019) Critical digital making (Knochel & Patton, 2015) Computational perspectives (questioning) (Brennan & Resnick, 2012) Computational creativity (Israel-Fishelson et al., 2021; Miller et al., 2014) CT in broader context of 'technological understanding' (Caeli & Bundsgaard, 2020) CT and digital competence (Juškevičiene & Dagiene, 2018) CT and MIL (Gretter & Yadav, 2016) CT and the #5c21framework (Romero et al., 2017)

in order to broaden participation in *computational communities* (Lachney, 2017). CT therefore should include interdisciplinary collaboration to incorporate discussion of relatable content and real-life applications (Tran, 2019) across different subjects (Barr & Stephenson, 2011).

The collaborative dimension overall emphasises that not only social and interpersonal skills are important, but also interdisciplinary communication and being proactively inclusive should be considered as fundamental learning opportunities within conceptualisations of CT.

Critical and Creative

The final dimension in our organisation of the literature is designated as critical and creative gathering literature that addresses the urgency of developing critical and creative thinking through coding (Table 4).

This dimension supports the idea that CT is not just about promoting individuals to be problem-solvers and interdisciplinary programmers, but to be creative thinkers and informed citizens for an inclusive society with technology (Iversen et al., 2018; Kafai et al., 2019; Lee & Soep, 2018). This has allowed some authors to find crossovers between CT and, for instance, UNESCO's MIL framework (Gretter & Yadav, 2016) or other approaches to digital literacy (Juškevičiene & Dagiene, 2018). The dynamic between creative and critical thinking is well described in Knochel and Patton's (2015) study where they highlight the value of engaging in computational thinking by playfully creating computer code as both an art medium and as an opportunity to critically think about the ways digital media impacts society. Throughout our sample we have identified different definitions of creativity and critical thinking leading to a range of understandings of how they combine with computational thinking.

Creativity is described in our sample as both an innovative way of approaching problem-solving and as a form of expression through the creation of artefacts (Brennan & Resnick, 2012). One way of conceptualising the merging of CT and creative thinking is through *computational creativity*: the idea that the more creative students are, the less time and effort it takes them to solve problems (Israel-Fishelson et al., 2021). Drawing from Epstein's Generativity Theory, Miller et al. (2014) identify four components of *computational creativity*: broadening knowledge, challenging established thinking, surrounding oneself with stimulus and capturing novelty. Nurturing these ways of thinking allows one to break through the wall of predetermined thoughts and apply concrete and innovative ideas to solve problems (Miller et al., 2014). For Korkmaz et al. (2017), creativity is about introducing new relations and combinations by looking at problems or events from different perspectives in order to find alternative solutions. As much CT is grounded in solution-seeking, for Michaelson (2015) is also about ensuring that equal weight is given to concrete information structures in real world problem scenarios. Overall, it is consensual in these studies that creativity should become more visible, but assessing creativity can be complex (Romero et al., 2017).

In terms of what critical thinking actually means in the context of CT, some papers define it as a methodological way of evaluating and making decisions in problem-solving (Berikan & Özdemir, 2020; Brennan & Resnick, 2012; Korkmaz et al., 2017; Voskoglou & Buckley, 2012). The latter is, for instance, described in Brennan and Resnick's (2012) *computational perspectives* (perceptions about individuals and the world around them), when the authors refer to the ability of *questioning* the taken-for-granted. The relationship of critical thinking and *questioning* is aligned with the analytical way of making conscious judgements, as described by the American Psychology Association (APA). Following this rationale, Korkmaz et al. (2017) define critical thinking in CT as the use of different methods to solve problems by effectively questioning one's decisions through evidence based proofs.

Critical thinking is also framed in a broader context of having a technological understanding of coding in society (Caeli & Bundsgaard, 2020; Juškevičiene & Dagiene, 2018), for instance, in the fostering of a responsible use of technologies by understanding the impact of computing in society that involves ethical issues and privacy concerns (Juškevičiene & Dagiene, 2018). Terminologies such as *critical computational thinking* (Kafai et al., 2019) or *critical computational literacy* (Lee & Soep, 2018) bring creativity and critical thinking together by emphasising both an understanding of power structures and production-oriented media literacy in order to develop culturally meaningful applications that have impact in the real world. Through this critical pedagogy approach, CT is seen as a way of engaging with the political, moral and ethical challenges of the world with concerns about social discrimination, such as racism or sexism (Kafai et al., 2019). It therefore places a deeper understanding of technology that goes beyond teaching the technical know-how, to value and nurture educational practices that centre on underrepresented societal minorities for inclusive participation in society (Lee & Soep, 2018). In line with this approach lies *computational empowerment* (CE) (Iversen et al., 2018), which argues that classic definitions of CT lack a critical and reflective stance towards a digitalized society. CE focuses on contextualisation and societal challenges, developing skills to engage diverse students for creative and critical participation with technology in society.

Critical and creative thinking are also highlighted in discourses about the 21st century skills as, for instance, in the #5c21 framework, in which Romero et al. (2017) propose an integrated approach to CT, with critical thought as the basis of all other 21st century skills (creativity, CT, problem-solving, and collaboration). Likewise, for Buitrago-Flórez et al. (2020), the 4Cs of 21st century skills (creativity, critical thinking, collaboration and communication) are crucial in the context of CT. Attempting to map these 21st century skills (creativity, critical thinking, and problem-solving) onto Brennan and Resnick's (2012) framework, Wong and Cheung (2020) found that CT interventions did enhance these skills, but caution is needed in assuming their 'transferability' to other domains. Overall, the critical and creative dimension highlights fundamental learning opportunities for an informed and inclusive engagement with coding in society.

Discussion

This review of the literature set out to find *what types of learning opportunities are revealed by a systematic mapping of the literature around CT conceptualisations.* Our intention is not to build a framework nor give an ultimate definition of CT, but rather "try to find similarities and relationships in the discussions about CT" (Voogt et al., 2015, p. 726) in order to enable more coherent pedagogical approaches overall.

In Table 5, we have mapped the identified learning opportunities found in the literature into the three dimensions: functional, collaborative, and critical and creative. This way of mapping makes visible *what else can be learned when coding*, showing the wide range of learning opportunities that can be nurtured through framing the teaching of coding. The learning opportunities identified in the functional dimension focus on the cognitive processes involved in formulating and solving problems with computers. In terms of coding, it can be readily seen how functional CT can help with learning the operational functioning of a programming language and its vocabulary (e.g., debugging). The learning opportunities synthesised in the collaborative dimension focus on interdisciplinary teamwork and participation. From this perspective, learning coding is also about sharing ideas, being inclusive, and engaging in online coding communities to find collective solutions to problems. Finally, the learning opportunities synthesised in the critical and creative dimension draw not only on logical thinking, but also on creativity "as

Table 5. Learning Opportunities Identified in the Configurative Literature Review.

Dimension	Learning opportunities	Description
Functional	Problem-solving	Learning how to identify, formulate and explore solutions to problems.
	Analytical thinking	Learning how to interrogate problems methodically.
	Algorithmic thinking	Developing logical reasoning.
	Decomposition	Learning to break problems down into smaller steps.
	Abstraction	Developing the ability to generalise solutions.
	Debugging	Learning to test and modify in order to refine solutions.
Collaborative	Collaboration	Learning to work in a team and find solutions together.
	Expressing	Learning how to share ideas in effective ways.
	Communication	Developing effective ways to share and present information to diverse audiences.
	Community participation	Learning to engage in existing communities of code and to build on other's work.
Critical and creative	Creative thinking	Learning to make novel interventions with coding and to find innovative solutions.
	Cultural production	Learning to use coding in a culturally responsive way with social justice awareness.
	Critical thinking	Developing ethical, political, and social understanding of code.
	Evaluation	Learning to make informed decisions.
	Questioning	Learning to question implications of code for society.
	Empowerment	Learning how to use code to voice opinions and mobilise towards social justice.

a tool to understand code, or to critique code and its usage" (Dufva & Dufva, 2016, p. 108). Through this lens, learning coding is about developing an informed understanding of the impact of coding in order to critically engage with bias and ethical implications of code in society. This approach to CT is increasingly important, given concerns around information bias in a fake news era ruled by algorithmic gatekeepers (Bell, 2014), and data transparency in a 'black box society' (Pasquale, 2015).

Whilst expanding conceptualisations of CT provides a comprehensive way of looking at the teaching of coding, it is also important not to over-simplify what we have found nor suggest static delineations. Firstly, so-called 'core' CT skills should be seen as overlapping and intertwined with wider transversal skills rather than fixed and isolated. Attempting to define CT 'core' competencies versus more 'peripheral' ones might create an unnecessary tension (Voogt et al., 2015). We find that using the term 'learning opportunities' is more inclusive, rather than the reductionist view implied by terms like 'components.' In addition, many definitions comprise varying mixtures of

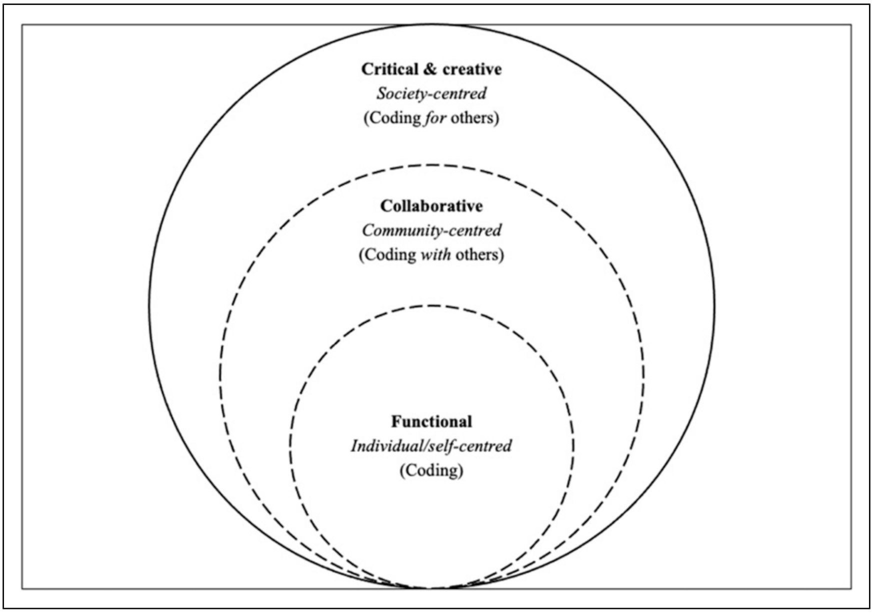

Figure 4. Visualisation of the three-dimensional mapping of CT literature.

specific and transversal competences, and in some cases attitudes, which are not exclusive to CT (Velázquez-Iturbide, 2018). Recognition that there remain difficulties in trying to bring CT into other research fields is one takeaway from this review. CT is still perceived as epistemologically challenging in non-CS fields, such as social sciences and humanities, suggesting the need for alternative design strategies (Czerkawski & Lyman, 2015).

Thirdly, awareness is needed that there are different understandings of the same term for a skill across the literature. For instance, the definition of critical in one paper may be different to the definition of critical in another paper. This can have direct implications in the way that skills are assessed and viewed as learning opportunities. In particular, there are different interpretations of creativity and critical thinking across the literature including how these two skills are related, hiding sometimes the interconnectedness at play. This might help explain why some reviews in our sample show a prominence of 'functional' skills being the most assessed (Cutumisu et al., 2019; Hasesk & Ilic, 2019), compared, for instance, to communication, collaboration, creativity, or critical thinking, which are complex to evaluate (Romero et al., 2017). Bringing awareness of different definitions is a starting point to make visible and stimulate discussion about the learning opportunities related to CT.

The three dimensions identified through the literature are related to each other in a dynamic and intertwined way. Our proposed organisational model as illustrated in Figure 4 seeks to highlight the ways in which the dimensions are built on top of each other in permeable layers of equal importance. This visualisation shows how the individual act of learning coding can be embedded in broader interactions with others and the world. For instance, problem-solving can be perceived both in terms of reading and writing code in a functional way, but also in terms of working in teams to find creative and critically informed solutions.

We also argue that the development of the learning opportunities can be understood at different levels: the functional dimension is more focussed on developing individual cognitive skills related to coding; the collaborative dimension is about coding *with* others in a community or a team; and the critical and creative dimension involves coding *for* others in the context of wider informed participation in society. This distinction is similar to what Brennan and Resnick (2012) describe as the value of creating *with* others, and the value of creating *for* others. We expand on the latter's values of being aware of a broader audience (society-at-large) when creating/coding *for* others.

This way of presenting different layers or dimensions related to CT is not a novelty. In our systematic analysis, Kafai et al. (2019) also use a circular model to present three epistemological framings of CT (*cognitive*, *situated*, and *critical*). Whilst there are undoubtedly similarities between both studies (as discussed in our findings), we focus on the learning opportunities of CT rather than the underpinning learning theories. For instance, while the *situated* frame is aligned with our *collaborative* dimension, it emphasises collaborative practices in the context of work, family and community, rather than developing the skills for interdisciplinary and intercultural dialogue in coding. One important crossover is the dynamics between the frames/dimensions illustrated in both circular models. For instance, Kafai et al. (2019) argue that the outer frames (*situated* and *critical*) are not progressions from the *cognitive* to better framings. Similarly in our model, we argue that the outer layers are grounded in the functional heart of CT, but learning should also be balanced across the dimensions.

Conclusion and Further Work

Given the importance of CT in the learning of coding, the learning opportunities revealed by organising the CT literature can help teachers and designers of coding courses. This is a significant contribution because whilst most courses for introducing coding to non-CS learners still centre on individual acquisition of operational aspects of programming (diSessa, 2018), our review suggests teaching coding through a broader range of learning opportunities that are meaningful to student lives and careers. For example, our collaborative dimension might point to interdisciplinary practices around the use of real-world examples, remixing other's work, and use of datasets that students can relate to and engage with meaningfully and critically. The pedagogic practices that support our multi-dimensional model have also been explored in further research, where we study how the learning opportunities in each dimension are fostered in practice (Tarling et al., 2022). If we want to provide meaningful learning to students, rather than just preparing them for the digital workforce (Dufva & Dufva, 2016), we need to educate them to communicate, collaborate, and question the ways in which coding, for instance, can ensure more under-represented voices contribute to the evolution of socio-technical practices (Baker-doyle, 2018; Ryoo et al., 2013).

Declaration of Conflicting Interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported by the Institute of Coding.

ORCID iD

Ana Melro https://orcid.org/0000-0002-5649-4957

Supplemental Material

Supplemental material for this article is available online.

Notes

1. Boolean search by topic (English language only): TS = ("Computational Thinking").
2. Similar to the way that 'functional literacy' was challenged by UNESCO, after its emergence in the 1950's, for being related to economic efficiency and productive workforce (Levine, 1982).

References

Baker-doyle, K. J. (2018). I, Pseudocoder: Reflections of a literacy teacher-educator on teaching coding as critical literacy. *Contemporary Issues in Technology and Teacher Education*, *18*(2), 255–270.

Barr, V., & Stephenson, C. (2011). Bringing computational thinking to K-12: What is involved and what is the role of the computer science education community? *ACM Inroads*, *2*(1), 48–54. https://doi.org/10.1145/1929887.1929905

Bell, E. (2014). *Silicon valley and journalism: Make up or break up*. Reuters memorial lecture. http://reutersinstitute.politics.ox.ac.uk/news/silicon-valley-and-journalism-make-or-break

Berikan, B., & Özdemir, S. (2020). Investigating "problem-solving with datasets" as an implementation of computational thinking: A literature review. *Journal of Educational Computing Research*, *58*(2), 502–534. https://doi.org/10.1177/0735633119845694

Bocconi, S., Chioccariello, A., Dettori, G., Ferrari, A., Engelhardt, K., Kampylis, P., & Punie, Y. (2016). Exploring the field of computational thinking as a 21st century skill. *Edulearn Proceedings*, *1*, 4725–4733. https://doi.org/10.21125/edulearn.2016.2136

Boyd, D., & Crawford, K. (2011). *Six provocations for big data*. Oxford Internet Institute - Symposium on the Dynamics of the Internet and Society. http://softwarestudies.com/cultural_analytics/Six_Provocations_for_Big_Data.pdf

Brennan, K., & Resnick, M. (2012). *New frameworks for studying and assessing the development of computational thinking*. American Educational Research Association Meeting (AERA). https://doi.org/10.1007/978-3-319-64051-8_9

Buitrago-Flórez, F., Danies, G., Tabima, J., Restrepo, S., & Hernández, C. (2020). Designing a socio-cultural approach for teaching and learning computational thinking. *Nordic Journal of Digital Literacy*, *15*(2), 106–124. https://doi.org/10.18261/issn.1891-943x-2020-02-03

Burke, Q., O'Byrne, W. I., & Kafai, Y. B. (2016). Computational participation. *Journal of Adolescent & Adult Literacy*, *59*(4), 371–375. https://doi.org/10.1002/jaal.496

Caeli, E. N., & Bundsgaard, J. (2020). Computational thinking in compulsory education: A survey study on initiatives and conceptions. *Educational Technology Research and Development*, *68*(1), 551–573. https://doi.org/10.1007/s11423-019-09694-z

Camp, T., Adrion, W. R., Bizot, B., Davidson, S., Hall, M., Hambrusch, S., Walker, E., & Zweben, S. (2017). Generation CS: The challenges of and responses to the enrollment surge. *ACM Inroads*, *8*(4), 59–65. https://doi.org/10.1145/3141773

Chilana, P. K., Alcock, C., Dembla, S., Ho, A., Hurst, A., Armstrong, B., & Guo, P. J. (2015). *Perceptions of non-CS majors in intro programming: The rise of the conversational programmer*. IEEE Computer Society. https://ieeexplore.ieee.org/document/7357224

Cuny, J., Snyder, L., & Wing, J. M. (2010). *Demystifying computational thinking for non-computer scientists*. The link magazine. http://www.cs.cmu.edu/~CompThink/resources/TheLinkWing.pdf

Cutumisu, M., Adams, C., & Lu, C. (2019). A scoping review of empirical research on recent computational thinking assessments. *Journal of Science Education and Technology*, *28*(6), 651–676. https://doi.org/10.1007/s10956-019-09799-3

Czerkawski, B. C., & Lyman, E. W. (2015). Exploring issues about computational thinking in higher education. *TechTrends*, *59*(2), 57–65. https://doi.org/10.1007/s11528-015-0840-3

Davenport, J. H., Crick, T., Hayes, A., & Hourizi, R. (2019). The Institute of Coding: Addressing the UK digital skills crisis. *ACM International Conference Proceeding Series*. https://doi.org/10.1145/3294016.3298736

Dawson, J. Q., Allen, M., & Campbell, A. (2018). Designing an introductory programming course to improve non-majors ' experiences. In SIGCSE'18: The 49th ACM Technical Symposium on Computer Science Education.

diSessa, A. A. (2001). *Changing minds: Computers, learning and literacy*. MIT Press.

diSessa, A. A. (2018). Computational literacy and "the big Picture" concerning computers in mathematics education. *Mathematical Thinking and Learning*, *20*(1), 3–31. https://doi.org/10.1080/10986065.2018.1403544

Dufva, T., & Dufva, M. (2016). Metaphors of code—structuring and broadening the discussion on teaching children to code. *Thinking Skills and Creativity*, *22*, 97–110. https://doi.org/10.1016/j.tsc.2016.09.004

Forte, A., & Guzdial, M. (2005). Motivation and nonmajors in computer science: Identifying discrete audiences for introductory courses. *IEEE Transactions on Education*, *48*(2), 248–253. https://doi.org/10.1109/TE.2004.842924

Gough, D., Oliver, S., & Thomas, J. (2017). *An introduction to systematic reviews*. Sage Publications Ltd. https://doi.org/10.1007/978-3-030-12263-8_1

Gretter, S., & Yadav, A. (2016). Computational thinking and media & information literacy: An integrated approach to teaching twenty-first century skills. *TechTrends*, *60*(5), 510–516. https://doi.org/10.1007/s11528-016-0098-4

Grover, S., & Pea, R. (2013). Computational thinking in K–12. *Educational Researcher*, *42*(1), 38–43. https://doi.org/10.3102/0013189X12463051

Grover, S., & Pea, R. (2017). Computational thinking: A competency whose time has come. Computer science education: perspectives on teaching and learning in school. http://hub.mspnet.org/index.cfm/33300

Hasesk, H. I., & Ilic, U. (2019). An investigation of the data collection instruments developed to measure computational thinking. *Informatics in Education*, *18*(2), 297–319. https://doi.org/10.15388/infedu.2019.14

Israel, M., Pearson, J. N., Tapia, T., Wherfel, Q. M., & Reese, G. (2015). Supporting all learners in school-wide computational thinking: A cross-case qualitative analysis. *Computers and Education*, *82*, 263–279. https://doi.org/10.1016/j.compedu.2014.11.022

Israel-Fishelson, R., Hershkovitz, A., Eguíluz, A., Garaizar, P., & Guenaga, M. (2021). The associations between computational thinking and creativity: The role of personal characteristics. *Journal of Educational Computing Research*, *58*(8), 1415–1447. https://doi.org/10.1177/0735633120940954

Iversen, O. S., Smith, R. C., & Dindler, C. (2018). *From computational thinking to computational empowerment*. Hasselt and Genk. https://doi.org/10.1145/3210586.3210592

Juškevičiene, A., & Dagiene, V. (2018). Computational thinking relationship with digital competence. *Informatics in Education*, *17*(2), 265–284. https://doi.org/10.15388/infedu.2018.14

Kafai, Y. (2016). From computational thinking to computational participation in K-12 education. *Communications of the ACM*, *59*(8), 26–27. https://doi.org/10.1145/2955114

Kafai, Y., Proctor, C., & Lui, D. (2019). From theory bias to theory dialogue. *International Computing Education Research Conference*, *11*(1), 44–53. https://doi.org/10.1145/3381887

Knochel, A. D., & Patton, R. M. (2015). If art education then critical digital making: Computational thinking and creative code. *Studies in Art Education*, *57*(1), 21–38. https://doi.org/10.1080/00393541.2015.11666280

Kong, S. C., Chiu, M. M., & Lai, M. (2018). A study of primary school students' interest, collaboration attitude, and programming empowerment in computational thinking education. *Computers and Education*, *127*, 178–189. https://doi.org/10.1016/j.compedu.2018.08.026

Korkmaz, Ö., Çakir, R., & Özden, M. Y. (2017). A validity and reliability study of the computational thinking scales (CTS). *Computers in Human Behavior*, *72*, 558–569. https://doi.org/10.1016/j.chb.2017.01.005

Lachney, M. (2017). Culturally responsive computing as brokerage: Toward asset building with education-based social movements. *Learning, Media and Technology*, *42*(4), 420–439. https://doi.org/10.1080/17439884.2016.1211679

Lee, C., & Soep, E. (2018). Beyond coding: Using critical computational literacy to transform tech. *Texas Education Review*, *6*(1), 10–16. https://doi.org/10.15781/T24J0BF37

Levine, K. (1982). *Functional literacy: Fond illusions and false economies*. Harvard educational review. https://doi.org/10.17763/haer.52.3.77p7168115610811

Lockwood, J., & Mooney, A. (2018). Computational thinking in secondary education: Where does it fit? A systematic literary review. *International Journal of Computer Science Education in Schools*, *2*(1), 41–60. https://doi.org/10.21585/ijcses.v2i1.26

Michaelson, G. (2015). Teaching programming with computational and informational thinking. *Journal of Pedagogic Development*, *5*(1), 51–66.

Miller, L. D., Soh, L. K., Chiriacescu, V., Ingraham, E., Shell, D. F., & Hazley, M. P. (2014). Integrating computational and creative thinking to improve learning and performance in CS1. In SIGCSE 2014 - Proceedings of the 45th ACM Technical Symposium on Computer Science Education. https://doi.org/10.1145/2538862.2538940

National Research Council. (2011). Report of a workshop on the pedagogical aspects of computational thinking. In *Report of a workshop on the pedagogical aspects of computational thinking*. https://doi.org/10.17226/13170

Orlik, J. (2019). *Five principles for innovation to prepare for a shifting world of work.*

Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. Basic Books.

Pasquale, F. (2015). *The black box society. The secret algorithms that control money and information*. Harvard University Press.

Pinkard, N., Martin, C. K., & Erete, S. (2020). Equitable approaches: Opportunities for computational thinking with emphasis on creative production and connections to community. *Interactive Learning Environments*, *28*(3), 347–361. https://doi.org/10.1080/10494820.2019.1636070

Popat, S., & Starkey, L. (2019). Learning to code or coding to learn? A systematic review. *Computers and Education*, *128*, 365–376. https://doi.org/10.1016/j.compedu.2018.10.005

Romero, M., Lepage, A., & Lille, B. (2017). Computational thinking development through creative programming in higher education. *International Journal of Educational Technology in Higher Education*, *14*(1), Article 42. https://doi.org/10.1186/s41239-017-0080-z

Ryoo, J. J., Margolis, J., Lee, C. H., Sandoval, C. D. M., & Goode, J. (2013). Democratizing computer science knowledge: Transforming the face of computer science through public high school education. *Learning, Media and Technology*, *38*(2), 161–181. https://doi.org/10.1080/17439884.2013.756514

Sax, L. J., Lehman, K. J., & Zavala, C. (2017). Examining the enrollment growth: Non-CS majors in CS1 courses. *SIGCSE 2017*, 513–518. https://doi.org/10.1145/3017680.3017781

Shute, V. J., Sun, C., & Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational Research Review*, *22*, 142–158. https://doi.org/10.1016/j.edurev.2017.09.003

Sondakh, D. E., Osman, K., & Zainudin, S. (2020). A Pilot study of an instrument to assess undergraduates' computational thinking Proficiency. *International Journal of Advanced*

Computer Science and Applications, *11*(11), 263–273. https://doi.org/10.14569/IJACSA.2020.0111134

Tang, K. Y., Chou, T. L., & Tsai, C. C. (2020a). A content analysis of computational thinking research: An international Publication trends and research typology. *Asia-Pacific Education Researcher*, *29*(1), 9–19. https://doi.org/10.1007/s40299-019-00442-8

Tang, X., Yin, Y., Lin, Q., Hadad, R., & Zhai, X. (2020b). Assessing computational thinking: A systematic review of empirical studies. *Computers and Education*, *148*, Article 103798. https://doi.org/10.1016/j.compedu.2019.103798

Tarling, G., Melro, A., Fujita, T., & Kleine Staarman, J. (2022). Making coding meaningful: University students' perceptions of bootcamp pedagogies. *Pedagogies: An International Journal*, *1*, Article 18. https://doi.org/10.1080/1554480x.2022.2077338

Taslibeyaz, E., Kursun, E., & Karaman, S. (2020). How to develop computational thinking: A systematic review of empirical studies. *Informatics in Education*, *19*(4), 701–719. https://doi.org/10.15388/INFEDU.2020.30

Tissenbaum, M., Sheldon, J., Seop, L., Lee, C. H., & Lao, N. (2017). Critical computational empowerment: Engaging youth as shapers of the digital future. In IEEE Global Engineering Education Conference, EDUCON. https://doi.org/10.1109/EDUCON.2017.7943078

Tran, Y. (2019). Computational thinking equity in elementary classrooms: What third-grade students know and can do. *Journal of Educational Computing Research*, *57*(1), 3–31. https://doi.org/10.1177/0735633117743918

Turner, C. (2019). *Teaching children coding is a waste of time, OECD chief says*. The Telegraph. https://www.telegraph.co.uk/education/2019/02/21/teaching-children-coding-waste-time-oecd-chief-says/

Vee, A. (2017). Understanding computer programming as a literacy. *Literacy in Composition Studies*, *1*(2), 42–64. https://doi.org/10.21623/1.1.2.4

Velázquez-Iturbide, J. Á. (2018). *Towards an analysis of computational thinking*. IEEE. https://doi.org/10.1145/1118178.1118215

Voogt, J., Fisser, P., Good, J., Mishra, P., & Yadav, A. (2015). Computational thinking in compulsory education: Towards an agenda for research and practice. *Education and Information Technologies*, *20*(4), 715–728. https://doi.org/10.1007/s10639-015-9412-6

Voskoglou, M. G., & Buckley, S. (2012). *Problem Solving and Computational Thinking in a Learning Environment*. *36*(4), 28–46.

Williamson, B. (2016). Coding the biodigital child: The biopolitics and pedagogic strategies of educational data science. *Pedagogy, Culture and Society*, *24*(3), 401–416. https://doi.org/10.1080/14681366.2016.1175499

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, *24*(3), 33–35. https://doi.org/10.1145/1118178.1118215

Wong, G. K. W., & Cheung, H. Y. (2020). Exploring children's perceptions of developing twenty-first century skills through computational thinking and programming. *Interactive Learning Environments*, *28*(4), 438–450. https://doi.org/10.1080/10494820.2018.1534245

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*(6), 565–568. https://doi.org/10.1007/s11528-016-0087-7

Yadav, A., Zhou, N., Mayfield, C., Hambrusch, S., & Korb, J. T. (2011). Introducing computational thinking in education courses. In Proceedings of the 42nd ACM Technical Symposium on Computer Science Education - SIGCSE. https://doi.org/10.1145/1953163.1953297

Author Biographies

Ana Melro is a Lecturer in Media and Communications at the University of Maia and a researcher in the Communication and Society Research Centre (University of Minho) and in the Technology and Intermedia Studies Research Centre (University of Maia), Portugal. She holds a European PhD in Media and Education Studies from University of Minho in partnership with the Autonomous University of Barcelona. She was a postdoctoral research fellow of the Institute of Coding, at the School of Education in the University of Exeter, and has been developing her research around media, technology, and young people. ORCID https://orcid.org/0000-0002-5649-4957

Georgie Tarling is a Lecturer in the School of Education at the University of Exeter. Before joining the university, she worked as a producer of factual television programmes and as a teacher of English, Film and Media in secondary schools and further education. She has also worked as a postdoctoral research fellow for the Institute of Coding and the South West Institute of Technology, Her research interests lie in conceptualizing and developing pedagogies which support criticality and creativity in relation to digital practices. She is involved in ongoing research with employers and course developers to design new curricula around data science at Further and Higher Education level. ORCID https://orcid.org/0000-0003-2699-005X

Taro Fujita is an Associate Professor in mathematics education in the School of Education at the University of Exeter. He was trained as a primary and secondary school teacher in Japan, and completed his PhD at the University of Southampton. Taro has worked in the field of mathematics education, first at Glasgow University (BEd) and then at Plymouth University (BEd and PGCE). His current research interests include the history of mathematics education, the teaching and learning of geometry in lower secondary schools, deductive reasoning and intuitive skills in geometry and the use of technology in mathematics education. ORCID https://orcid.org/0000-0002-3547-456X

Judith Kleine Staarman is a Senior Lecturer in Education the School of Education at the University of Exeter, Programme Leader of the MA Technology, Creativity and Thinking and Director of Thinking Schools @Exeter. She completed her PhD at the Radboud University Nijmegen, The Netherlands. Prior to her current position at the University of Exeter, she worked as a researcher and lecturer at the Radboud University, The Open University and the University of Cambridge. She is interested in the process by which people learn, think and engage in creative processes together through collaboration and dialogue, both with and without technology. She also has a wider interest in the role of technology for teaching, learning and thinking, in particular in relation to education futures and how technology may change these possible futures. She has been involved in a range of research projects, focusing on the role of technology and dialogue in teaching and learning processes and has published widely on these topics. ORCID https://orcid.org/0000-0002-9504-1918