

JYX



JYVÄSKYLÄN YLIOPISTO
UNIVERSITY OF JYVÄSKYLÄ

This is a self-archived version of an original article. This version may differ from the original in pagination and typographic details.

Author(s): Mertala, Pekka

Title: The pedagogy of multiliteracies as a code breaker : A suggestion for a transversal approach to computing education in basic education

Year: 2021

Version: Published version

Copyright: © 2021 The Author. British Journal of Educational Technology published by John V

Rights: CC BY-NC 4.0

Rights url: <https://creativecommons.org/licenses/by-nc/4.0/>

Please cite the original version:

Mertala, P. (2021). The pedagogy of multiliteracies as a code breaker : A suggestion for a transversal approach to computing education in basic education. British Journal of Educational Technology, 52(6), 2227-2241. <https://doi.org/10.1111/bjet.13125>

The pedagogy of multiliteracies as a code breaker: A suggestion for a transversal approach to computing education in basic education

Pekka Mertala 

Faculty of Education and Psychology,
University of Jyväskylä, Jyväskylä, Finland

Correspondence

Pekka Mertala, Faculty of Education and
Psychology, University of Jyväskylä, Alvar
Aallon katu 9, 40600 Jyväskylä, Finland.
Email: pekka.o.mertala@jyu.fi

Funding information

None

Abstract

While computing has been (re)introduced into the basic education curricula in various countries, its actual implementation appears to be inconsistent. There are schools in which computing education is commonplace, while the implementation seems to be lagging behind in others. There is emerging evidence that some teachers do not consider computing education relevant, meaningful and important and, thus, intentionally neglect its provision. This is problematic as understanding the principles of code and computing is crucial for agentic citizenship in the post-digital era. This paper argues that one main reason for these teachers' reluctance is the economy-driven discursive framing of computing education, which is in contrast with the socialization-oriented manner in which teachers approach their work. To contribute to resolving this issue, the present paper introduces a transversal approach to computing education. It conceptualizes code as a sociomaterial text with social and societal histories and consequences. Theoretically and conceptually, the approach draws

This is an open access article under the terms of the Creative Commons Attribution-NonCommercial License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited and is not used for commercial purposes.

© 2021 The Author. *British Journal of Educational Technology* published by John Wiley & Sons Ltd on behalf of British Educational Research Association

on the pedagogy of multiliteracies. The leading idea is that digital technologies are examined with students from functional and critical dimensions and through micro and macro perspectives. The use of wearable sports technologies, such as activity wristbands, are used as practical examples to put the theoretical ideas into context.

KEYWORDS

algorithm, coding, computing, education, multiliteracies, pedagogy, school

Practitioner notes

What is already known about this topic

- Computing has been (re)introduced in the curricula of basic education in various countries.
- Some teachers are found to be reluctant to teach computing in basic education.

What this paper adds

- This paper introduces a transversal multiliteracies-based approach for computing education.

Implications for practice and policy

- Computing should be included in curricula and classrooms in a holistic manner that includes both functional and critical approaches to computing.

INTRODUCTION

While computing has been (re)introduced into the basic education curricula in various countries (Bresnihan et al., 2015; McGarr & Johnston, 2020; Mertala et al., 2020; Sentance & Csizmadia, 2017; Williamson et al., 2019), its actual implementation appears to be inconsistent. There are schools in which computing education is commonplace (Duncan et al., 2017; Geldreich et al., 2018), while the implementation seems to be lagging behind in others (Larke, 2019; Tanhua-Piironen et al., 2020). As a result, significant public (eg, Dickens, 2016; McDonald, 2017) and scholarly (eg, Duncan et al., 2017; Mason & Rich, 2019; Rich et al., 2021; Sentance & Csizmadia, 2017) debate has focused on the question of how to ensure that all teachers are qualified and competent enough to teach computing.

The situation, we argue, is more complex. First, it is worth questioning whether the mismatch between curricular alignments and educational praxis is solely due to actual or perceived lack of competence, as teachers have successfully implemented computing education while possessing only rudimentary skills (Duncan et al., 2017). There is emerging evidence that some teachers—at least in the context of primary education—do not consider computing education relevant or meaningful in the first place and therefore intentionally neglect its provision (Larke, 2019). We argue that this situation has much to do with the way in which computing and its educational objectives are introduced into contemporary curricula. The (re)introduction of computing education is often framed with economy- and employment-based

rationales and objectives (Bresnihan et al., 2015; Mertala et al., 2020; Williamson et al., 2019), which overshadow the more student-centred goals that have guided the development of computing education since the 1970s (eg, Solomon et al., 2020). Economy-driven discourses also contrast with the manner in which teachers approach their work: instead of training students to be future workers, teachers believe that their task as educators is to help students become agentic and independent citizens (Mertala, 2019b) who 'can understand what is happening and make informed choices, as members of a computationally-steeped democracy', to quote Wardrip-Fruin's (2015, p. 10) reformulation of Ted Nelson's famous argument on why we must understand computers. Indeed, teachers in Mee's (2020, p. 3) survey expressed concern that 'the current pressure to focus on programming and coding is already resulting in a decline of wider digital competencies' required in a digitalized society, and primary teachers in Larke's (2019) study used their professional judgement to modify or reject England's National Curriculum on computing standards by minimizing or ignoring subject content that they deemed redundant or less than critical to their students' success. Second, the notions above imply that even an active implementation of computing education does not necessarily equate to a pedagogy that supports children's agentic subjectivity in a computationally steeped society, and a review of existing research (eg, Duncan et al., 2017; Fagerlund et al., 2021; Geldreich et al., 2018; Mertala et al., 2020; Otterborn et al., 2020; Papadakis & Kalogiannakis, 2020; Sáez-López et al., 2016; Rich et al., 2019; Vega & Cañas, 2019) suggests that technical and functional aspects are dominant in computing education and that societal issues are touched upon in only a limited manner at best.

To overcome these obstacles, this conceptual paper proposes a more comprehensive approach for computing education. This so-called transversal computing education draws on the pedagogy of multiliteracies (The New London Group [NLG], 1996) and conceptualizes code as a sociomaterial text. The benefits of this conceptualization are twofold. First, by allowing us to draw on the rich and rigorous theoretical and conceptual resources of literacy research, it broadens the perspective of computing education from a purely technical level to acknowledge the social and societal issues related to code and coding (see also Vee, 2017). Second, what has been referred to as computing literacy overlaps with concepts such as data literacy (Pangrazio & Selwyn, 2020), digital literacy (Pangrazio & Sefton-Green, 2021) and media literacy (Valtonen et al., 2019). Thus, using an inclusive literacy framework enables a joint exploration of the assemblage-like nature of code, data and other related phenomena, instead of building conceptual walls between them.

Before moving further, a few conceptual remarks need to be made as the terminology around computing education is diverse (Manches & Plowman, 2017; Rich et al., 2021). In this paper, we follow Berry's (2013) definition of computing in an educational context as being 'concerned with how computers and computer systems work and how they are designed and programmed'—a description often used in scholarly debates on computing education in the childhood era (Manches & Plowman, 2017). By emphasizing programming and the question of how computers work, this definition places code at the centre of computing education, which is a view shared by many scholars approaching computing from a societal perspective (eg, Dufva & Dufva, 2016; O'Neil, 2016; Vee, 2017; Williamson, 2016). Thus, the present paper puts more emphasis on the digital realm of computing than those that approach it from the viewpoint of computational (Wing, 2006) or algorithmic (Futschek, 2006) thinking, for instance.

COMPUTING IN CURRICULA: TENSIONED RELATIONSHIP BETWEEN FUNCTIONAL AND CRITICAL APPROACHES

Curricula are not just 'out there' waiting to be discovered (Gordon, 2009); they are imagined and constructed by the agents involved in the process either directly or indirectly. Because

different agents possess different interests, the curriculum can take various forms depending on whose knowledge and values are included (Apple, 1990). In computing education, a rough division between functional and critical paradigms of computing can be made.

The functional paradigm emphasizes the logical aspects of computing, and it is often conceptualized under concepts such as algorithmic thinking (Futschek, 2006), computational thinking (Wing, 2006), coding skills (Tuomi et al., 2018), and computing literacy (Vee, 2013). Historically, the roots of functional computing education are typically located in Papert's classical work on Logo (Manches & Plowman, 2017; Solomon et al., 2020). For Papert, learning to program was not a goal in itself, but Logo was designed as a tool for more generic explorative learning, especially in mathematics (Solomon et al., 2020). Another influential work is Wing's (2006) paper on computational thinking—a collection of mental tools, such as decomposition, abstraction and heuristic reasoning, that reflect the breadth of the field of computer science—which she describes as a fundamental skill for everyone and comparable with the 'three Rs': reading, writing and arithmetic.

Functional computing education typically starts with physical manipulatives, such as Bee-Bot and other programmable floor robots (Papadakis & Kalogiannakis, 2020), or unplugged activities, including creating and following symbol or verbal instructions (Otterborn et al., 2020). Later, graphical programming interfaces and text-based languages are introduced (Sáez-López et al., 2016; Vega & Cañas, 2019). As these examples illustrate, in the functional paradigm, code is seen as a sequential set of instructions that are input into and processed by a machine (Dufva & Dufva, 2016)—a notion neatly captured in Vee's (2013) definition of computational literacy as the following:

... the constellation of abilities to break a complex process down into small procedures and then express—or 'write'—those procedures using the technology of code that may be 'read' by a non-human entity such as a computer. (n.p.)

The above definition also provides important cues about the two scientific fields the functional approach is found within: cognitive psychology (the ability to break a complex process down into small procedures) and computer science (to express the procedure in a form that enables the used computer's computing power)—both identifiable in Papert's (1980) and Wing's (2006) works. This disciplinary foundation defines the perspectives from which code and computing are observed and discussed. As a result, functional computing education pays only little attention to societal-level ethical questions concerning code. For example, in their highly cited paper, Brennan and Resnick (2012, p. 8) describe critical code-reading capacities as an understanding of what is reasonable to borrow from others and how to give appropriate credit to others. While both are ethical questions, they stay closely within the realm of computer science.

During recent years, code and programming have attracted the curiosity of scholars working in the critical branches of social sciences, and questions regarding code's societal consequences and how social reality is reflected in code have become a topical theme in academic discussions (eg, Barassi, 2020; Dufva & Dufva, 2016; Hobbs, 2020; Rantala, 2018; Williamson, 2016, 2017). In these writings, code is approached as a sociomaterial text either explicitly (Mertala et al., 2020; Vee, 2017) or implicitly (eg, Barassi, 2020; O'Neil, 2016). The following quotation from Williamson (2016, p. 49) includes two important cues about the benefits of the approach:

Programming code captures ideas about how the world works and translates them into formalized models that can be computed through algorithmic procedures, which can then augment, mediate and regulate people's lives.

First, the idea that ‘code captures ideas about how the world works’ (Williamson, 2016, p. 49) suggests that code in digital technologies is not value-free but that it ‘widely reflects both conscious and subliminal values of the programmer, a software company or society’s understanding of good code’ (Dufva & Dufva, 2016, p. 98). Second, by highlighting that code can ‘augment, mediate and regulate people’s lives’ (Williamson, 2016, p. 49), the quotation argues that code is a type of text that is used to create digital artefacts, such as different applications, and that via these applications, code has social and societal consequences.

O’Neil (2016) provides various concrete cases to put these rather abstract arguments into context by describing how automated algorithm-driven solutions reflect, maintain and even enforce societal inequality. For example, crime detection algorithms used in the United States are not effective in detecting ‘white collar’ financial frauds, such as tax evasion. Instead, they are relatively effective in detecting burglaries and minor drug trafficking often conducted by people from less fortunate backgrounds. The more data are collected via arrests, the more precise and effective—but also more specialized—these algorithms become, and the ‘result is that we criminalize poverty, believing all the while that our tools are not only scientific but also fair’ (O’Neil, 2016, p. 91; see also Barassi, 2020).

While critical viewpoints have gained a foothold in scholarly contexts, they are not yet widely manifested in mainstream curricula. Instead, the functional view of code and computing dominates in curricula globally (eg, Williamson et al., 2019; Wu et al., 2020). One possible explanation for the paucity of critical approaches is that studies carried out in various countries, including Australia, England, Finland, Ireland and Sweden, have identified that the interests of the technology industry have played a notable role in shaping the way the ‘whys’ and ‘hows’ of computing have been attached to curricula (Bresnihan et al., 2015; McGarr & Johnston, 2020; Mertala et al., 2020; Williamson et al., 2019). Coding skills are claimed to be ‘a success factor for a society’ (Tuomi et al., 2018, p. 419), and the main objective for computing education—alongside supporting students’ problem solving and learning skills—appears to be to maximize the potential pool of future coders and tech entrepreneurs (Larke, 2019; Mertala et al., 2020; Williamson, 2017; Williamson et al., 2019).

That being said, it is important to acknowledge that exceptions, despite being smaller in scale, exist. One example of an alternative curriculum is *The Beauty and Joy of Computing* (BJC) by the University of Berkley, which pays attention to the societal impacts of code by reminding students ‘that the decisions about how a new technology is used are made by human beings, including themselves if they pursue a career in computer science, so they shouldn’t feel helpless in the face of a supposed technological imperative’ (BJC, n.d.). As another example, Lee and Soep (2016) introduced the idea of ‘critical computing literacy’, which brings together critical literacy and computational thinking, and tested this idea in a classroom context—a topic to be discussed in more detail in the following section.

COMPUTING AS PART OF A BROADER LITERACY FRAMEWORK

We are not the first to suggest that instead of teaching computing as a separate domain, it could be included in some existing literacy frameworks. Valtonen et al., (2019) recently suggested that computing education could be located under media literacy education because algorithms and automation have taken over a notable amount of media processes. Others (eg, Campbell & Walsh, 2017; Pangrazio & Sefton-Green, 2021) locate coding within a digital literacy framework among a variety of other skills. While such approaches can offer a base for a more holistic pedagogical take on code and computing, they are not without problems. Let us first examine Lee and Soep’s (2016) critical computing literacy. In their paper, Lee and Soep (2016) report on a project that combined critical media literacy and computer programming. In the project, high school students created counter-stories to their existing

dominant narratives of urban youths of colour by making digital games via Scratch. Although the idea of introducing game design and coding as a form of agential self-expression is respectable, it seems that the critical aspect of the approach does not touch upon that much code and computing. Although the project unquestionably shows the students that coding can be used to create media texts that criticize and challenge the dominant narratives, the take on code and coding per se remains at a rather functional level, and the political and persuasive dimensions of code remain untouched.

One explanation for this dichotomy is that both media literacy and functional computing education have long histories, strong identities and specialized vocabularies, which can make synthesis a challenging process. When it comes to situating computing education under digital literacy, an increasing number of scholars have begun to describe our world as post-digital (eg, Berry, 2014; Dufva & Dufva, 2019; Jandric et al., 2018). The concept 'post-digital' proposes that as 'the digital has become completely bound up with and constitutive of everyday life' (Berry, 2014, p. 15), using 'digital' as a defining concept fails to capture the essence of our era.

One more issue is that regardless of whether the approach to computing is solely functional or located within media literacy or digital literacy, code is typically observed and discussed as a phenomenon detached from school. For example, Valtonen et al. (2019) otherwise detailed account of technologies of media literacy does not touch upon how many of the listed technologies (eg, tracking, recommenders and optimization) operate in schools as well as in the form of learning management systems, learning analytics and artificial intelligence-based facial recognition systems (Selwyn, 2019). These technologies, to paraphrase the previous quote from Williamson (2016), augment, mediate and regulate students' actions and choices in a roughly similar manner as technologies used outside school. This view is well captured in Selwyn's (2019, p. 69) argument that 'any intelligent tutoring system or pedagogical agent is essentially a form of individually focused behaviour management' as the argument's content and terminology are notably in parallel with the idea of personalized behaviour engineering often used to summarize logics of commercial digital platforms (Valtonen et al., 2019).

To overcome the above-mentioned issues, we suggest that pedagogy multiliteracies can provide conceptual and pedagogical foundation for transversal computing education, which combines functional and critical paradigms. The idea of the pedagogy of multiliteracies was coined by NLG in the mid-1990s. NLG defined multiliteracies as a pedagogical approach that is required to meet the needs of the ever-diversifying textual and cultural landscapes of contemporary societies (NLG, 1996). Unlike in media literacy and digital literacy, in the pedagogy of multiliteracies, literacy is not an educational outcome and/or competence. Instead, multiliteracies are described as a form of pedagogy, that is, a 'teaching and learning relationship that creates the potential for building learning conditions leading to full and equitable societal participation' (NLG, 1996, p. 60).

Full and equitable societal participation is not a matter of course, and we understand this objective to be parallel with the individualistic dimension of the socialization task of education. In the traditional view of socialization, the role of education is to train students to be functional members of society 'as it is' (Biesta et al., 2015)—a view that is present in the current work-life-oriented rationales behind functional computing education. The individualistic dimension, in turn, encourages students to criticize the prevalent societal structures and to act as agents of change as they contribute to the development of a society that 'might be' (Biesta et al., 2015). To put this in context, how the post-digital world is perceived—as a given or as something that is produced and, thus, can be shaped—determines what kinds of futures are thought to be possible (Dufva & Dufva, 2019). NLG (1996) refers to this idea by stating that education can 'instantiate a vision through pedagogy that creates in microcosm a transformed set of relationships and possibilities for social futures, a vision that is lived

in schools' (p. 72). Put differently, the vision of the society that 'might be' is something that needs to be actualized in the everyday praxis of education.

While such a view is evidently idealistic, it serves as an important reminder that many of the naturalized and taken-for-granted traditions of school education can—and need to—be put under critical evaluation. Computing-wise, the idea of school as a microcosm demands that the pervasive mechanisms of schools' computational practices are made visible to the students. To draw on the concepts of NLG (1996), a self-critical gaze towards algorithmic school pedagogies can be conceptualized as the critical framing of situated practices. The core idea here is that education needs to be grounded in students' everyday experiences of code both in and out of school (the micro level); however, they must also go beyond these by making visible how they relate to the role of code in society (the macro level). In the following sections, we provide an overview of how transversal computing education could be practiced in basic education.

TOWARDS TRANSVERSAL COMPUTING EDUCATION

Even though we criticized the functional approach to computing in the previous sections, we are not advocating rejecting it. Instead, as any singular view of code is not sufficient in itself (Dufva & Dufva, 2016), we wish to highlight the importance of supplementing functional computing education with critical perspectives, as the combination of these will arguably support and promote students' agentic subjectivity in a post-digital society, rather than relying on only one dimension. An illustrative example of the interrelatedness of functional and critical perspectives is artist Simon Weckert's traffic jam installation (see Barrett, 2020), which he created by towing a cart loaded with 99 cellphones up and down a street in central Berlin. All of the phones had their location services turned on. As a result, Google Maps began to warn drivers about a major traffic jam on the street Weckert had walked and suggested alternative routes, which the drivers apparently followed. According to Weckert, the motive behind the installation was to illustrate how by making changes to the digital world, one can alter the physical world. Thus, even though Weckert's objectives were critical, creating the installation would not have been possible without a functional understanding of how the algorithms of Google Maps work. The general idea of transversal computing education is summarized in Figure 1 and discussed in more detail below.

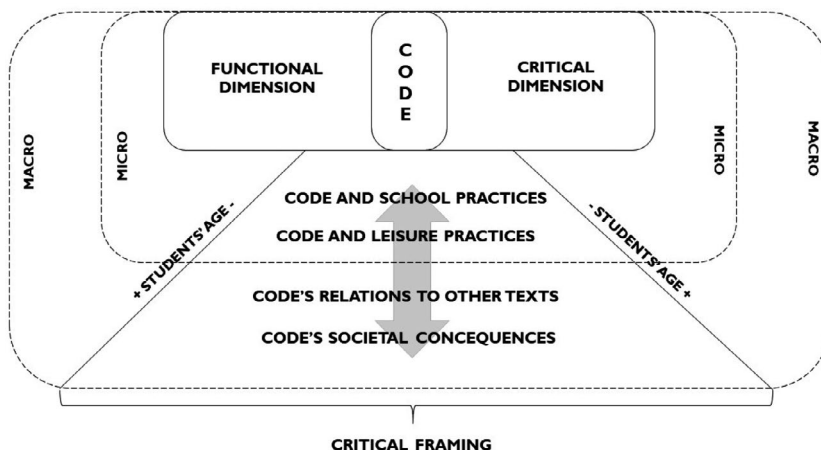


FIGURE 1 Transversal computing education

The leading idea of transversal computing education in basic education is the following: computational technologies used in school and leisure time are examined with the students (a) from functional and critical dimensions and (b) by switching between the often overlapping micro- and macro-level perspectives. For example, by exploring logics of YouTube (code and students' leisure practices), it is possible to display how users' viewing histories affect the kinds of videos they are recommended in the future (code's relations to other texts, namely, data). This notion serves as a bridge for a more general discussion on how algorithmic personalization filters the information we are provided with and, thus, plays a role in our commercial and political choices and actions. That said, to avoid an over-deterministic tone, it is equally important to pay attention to the blind spots and restrictions of code. Topics to be discussed could include whether YouTube's algorithm can detect if the same account is used by different family members or if the algorithm can 'understand' one's motivation for watching a particular video. Similar observations can then be placed on school technologies: can learning analytics software tell whether you are providing inaccurate answers due to a lack of knowledge or just to play the fool?

We acknowledge that the above-discussed examples take place on rather an abstract level and, thus, are more suitable to explore with older students. With the youngest of students, transversal computing education will focus on the micro level: concrete issues and immediate experiences. To provide a practical example, we (see Mertala et al., 2020) tested how the blind spots of accelerometer-informed algorithms used in activity wristbands can be illustrated for children aged 4–6 years by controlling the movements of their hands: strong waving of their hands while sitting on the floor was counted as steps, whereas walking with hands in a fixed position was not. These observations were then hypothesized with the children, and the hypotheses were tested. In the following sections, we dig deeper into the various ways these wearable sports technologies (WSTs) can be used to scrutinize the interplay between the micro and macro levels and functional and critical dimensions.

Quantified corporals: WSTs and physical education

WSTs are an information-rich case for several reasons. First, (young) students seldom recognize and/or conceptualize wearables as a form of ubiquitous computing (Mertala, 2019a). Thus, it is important to broaden their understanding about the scope of the algorithm-driven technologies with which they interact. Second, wearables are commonly used in basic education. Concerns related to children's and adolescents' obesity and their lack of physical exercise are a common theme in public discussions (Merikivi et al., 2016), and the use of digital technologies in physical education is seen as one main solution for the problem. Typically, this means using wearable and/or tracking devices, such as heart rate monitors and activity wristbands, in physical education classes as well as analysing the data collected by these devices (eg, Koivisto et al., 2017; Lupton, 2021; Mikkola et al., 2011; Williamson, 2015). Lastly, WSTs share the same basic logics and characters with many other forms of technologies students encounter both in and out of school: like other mobile devices, GPS-enabled WST collect geo-locational data from the users (Pangrazio & Selwyn, 2019). In addition, WST can be understood as persuasive technologies, whereby algorithms are designed to shape the user's behaviour in a similar manner as algorithmic recommendation systems (Valtonen et al., 2019) and pedagogical agents of learning analytics software (Selwyn, 2019).

To summarize, WST can be conceptualized as surveillance technologies in which datafication and algorithms form an assemblage that cannot be grasped by only focusing on one text (ie, data or code). Table 1 compresses the questions, themes and phenomena that can be approached and made visible through critical framing. The contents of each of the four fields are discussed in more detail after the table.

TABLE 1 Examples of how WST can be utilized in transversal computing education

	Functional dimension	Critical dimension
Micro level	Exploration of the technical features of the hardware and software used in WST <ul style="list-style-type: none"> • What does it measure and how? • What does it analyse and how? • What does it report and how? • What kind of code are WSTs built on? 	Analysis of the data double ^a and analyses produced by the algorithmic technology <ul style="list-style-type: none"> • What background information is collected from the user? • Why are particular variables measured and analysed instead of others?
Macro level	Planning how the identified shortcomings of WST can be adjusted by improving the code <ul style="list-style-type: none"> • What kind of functional computing skills does the improvement require? • What kind of limitations do the computational models of external and internal worlds convey? 	Examination of WST as an example of a broader category of tracking and surveillance technologies <ul style="list-style-type: none"> • Illustrating how other products and services (devices and software) follow similar algorithmic logic as WST. • Exploring tracking technologies as a form of external governance (eg, workforce monitoring).

^aA decorporalized virtual version of an individual produced by and constructed from a series of data flows (Haggerty & Ericson, 2000).

Critical framing and micro-level phenomena

From a functional perspective, the foremost objective is to explore the kinds of data the device is able to capture, as well as how it captures and analyses the data. This is an important task as no measurement technology is flawless, and research has identified a number of factors that impair the accuracy of the measurement and analysis of WST (van der Kruk & Reijne, 2018). WST have been called sorting systems that highlight certain forms of movement and ignore others (Williamson, 2015, p. 140) by being selective with regard to the variables from the collected data indicating physical activity. Put differently, WST emphasize variables that can be measured and turned into numerical data, while simultaneously neglecting other variables that are as (or even more) important but more difficult to measure (Sharon & Zandenbergen, 2017). Take walking, for example. Research suggests that walking pace (the intensity of the steps) is often a better indicator of physical activity than the number of steps (Tudor-Locke et al., 2019). However, the number of steps has remained the main indicator of physical activity, an illustrative example of which is the enduring popularity of the 10,000-step rule (Vandelanotte et al., 2020), despite the fact that the number has limited scientific basis (Lee et al., 2019).

From a critical viewpoint, one question worth pondering with the students is why the developers have ended up measuring and valuing the number of steps instead of the intensity of steps. One explanation would be that the number of steps is a more straightforward indicator than the intensity of steps. Another explanation is that technology-wise, it is easier to capture and compare the number of steps taken, rather than the intensity of steps, as the latter would require data about pace, cadence, and heart-rate-levels. Easiness, however, does not necessarily correlate with accuracy. Accelerometer-based activity wristbands tend to consider large, continuous arm movements as steps, while simultaneously neglecting forms of physical activity in which the hands are static (eg, riding a bicycle), making accelerometer-informed algorithmic devices a rather unreliable technology to detect, analyse and evaluate physical movement (Chen et al., 2016).

Inaccuracy is also present in other forms of sensor data. For example, humans do not contain a universal serial bus (USB) port that provides accurate information about our heart rate. Instead, these data are only averages and algorithmically generated statistical data based on different indicators, such as vasodilation. Furthermore, the lack of a USB port

also means that the device is unable to read the user's physical condition or state of health. Thus, the device relies on proxies such as age, gender, height, weight and self-estimation of one's general physical activity as the basis for profiling. From a critical perspective, it is crucial to critically reflect whether humans can be reduced to a half-dozen variables and highlight the rather fundamental differences between the student subject and their 'data double' (Haggerty & Ericson, 2000), which serves as the basis of algorithm-driven analysis. For instance, the data double does not get sick nor does it suffer from knee pain.

Critical framing and macro-level phenomena

From a functional point of view, it is important to consider how the code could be improved to take better account of the complexities of the outside world and the internal world of the human subject. For the latter, an exemplary case to discuss is the activity–rest balance. Low-cost WST, for example, typically have fixed scales for daily activity (eg, Polar, n.d.), which provide little to no help in designing well-balanced training programmes. That said, it should be noted that more high-end WST provide the user with information about the physical load of each exercise and calculate recovery times based on various indicators, including heart rate data. Such an example serves as a practical example of conditional constructs: *if* condition *X* is met (ie, data points indicating a certain level of physical load is achieved), *then* the programme performs function *Y* (recommends activities to the user that are designed to stay under a certain level of physical load). The accuracy of many high(er)-end wrist-worn meters, however, is activity-dependent, as the measurement errors are higher in cycling than in running (Gillinov et al., 2017). This means that similar loads from different activities can provide different results and recommendations. Thus, if we have an idea how certain activities distort the data, then we can write an algorithm that fixes the error: *if* the user chooses cycling as the activity type, *then* the algorithm should add or reduce a certain coefficient when estimating the physical load. While the actual (functional) writing of *if*–*then* constructs can be practiced with more simple cases, thought exercises of this kind are appropriate for illustrating the kinds of measuring errors and calculations that take place underneath the interface of WST. This notion serves as a bridge to the critical dimension.

Even though the shortcomings of the measurement technology can be partially fixed by improving the algorithm, it is worth questioning whether the 'fix' in the code would work for all users and in all circumstances. Not all users are alike, and the wearer's skin colour, the ambient light and the type and degree of motion can affect the quality of the signal of wrist-worn optical meters (Parak, 2018, p. 16). Additionally, the idea of fixing the code is based on an assumption that the measurement error is a static phenomenon. This, however, is not the case as the accuracy of wrist-worn devices tends to decrease as intensity increases (Pasadyn et al., 2019), which means that the most loading exercises provide the most inaccurate data—a notion that is highly important from the viewpoint of the activity–rest balance. Additionally, it is important to highlight that similar logics are present in other algorithmic technologies as well. Take the content recommendation systems discussed earlier in the paper, for example: *if* one has watched *A* and *B* but not *C* and *D* from Netflix, *then* the algorithm recommends one to watch *C* and *D* *if* others who have watched *A* and *B* have also watched *C* and *D*. Furthermore, *if* the ones who watched *A*, *B*, *C* and *D* have rated *D* over *C*, *then* the algorithm is more likely to recommend *D* as the first choice (see also Valtonen et al., 2019). While such an example is inevitably simplified—which needs to be clarified for the students as well—thought exercises like this are valuable for understanding the logics, similarities and differences of various algorithmic technologies.

Lastly, WST—as one form of tracking technologies—offer a platform to discuss the role of algorithm-driven databased surveillance in society at large. As noted at the beginning of this

paper, the impetus for the (re)introduction of computing education is grounded on economy-based rationales and objectives. However, the potential economic benefits will most likely be unevenly distributed and adhere to (and confirm) the prevailing positions of power and society. Examples of how tracking technologies have become tools of governance between employers and employees are easy to find. With a large portion of the working population working from home due to Covid-19, employers are concerned about workforce productivity and the security of their business's confidential data. To combat these concerns, some employers are increasing digital employee monitoring (Osborne Clarke, 2020). Additionally, Amazon, the world's largest online store, automatically monitors the efficiency of packaging workers by measuring the time off task (ToT) value. If the ToT value exceeds the 30-min limit, the employee receives an automatic warning from the application. The supervisor then demands an explanation from the employee for the absence. If the explanation is acceptable, the supervisor may override the warning. If the ToT value exceeds 2 hr, the employee is fired (Lecher, 2019).

Pedagogical self-critique

As the last point, we want to emphasize the importance of pedagogical self-critique, by which we mean that teachers must approach their own use of algorithmic technologies with a similar critical attitude they teach to their students. Indeed, it would be rather hypocritical to teach students about the shortcomings of WST and, at the same time, rely uncritically on WST when evaluating students' progress (or the lack of) in physical education. This naturally applies to all forms of automated assessment and evaluation. Furthermore, it would also be paradoxical to use critical framing for WST but introduce other algorithmic technologies, such as learning analytics or the facial recognition used in video conference software, as 'value free', as they share the very same limitations and problems as WST. As previously mentioned, skin colour can affect the accuracy of optometric heart rate monitoring (Parak, 2018). Likewise, the facial recognition algorithms of various video conference software appear to adjust better for light- than dark-skinned users, which makes it more difficult for a dark-skinned student to use virtual backgrounds (Dickey, 2020). Via these examples, we wish to highlight that transversal computing education is not something that can be fully conducted as a 1-week project or as weekly lessons. While occasional intensive sessions can be beneficial, transversal computing education generally takes place in everyday interactions between students, teachers and algorithmic technologies.

CONCLUDING REMARKS

This article has introduced the idea of transversal computing education, which draws from the pedagogy of multiliteracies (NLG, 1996). The suggested approach moves beyond the now-prevalent functional paradigm by approaching code as sociomaterial text with societal roots and consequences. Highlighting the importance of this so-called critical dimension would not mean undermining the need for functional computing education; in fact, it is worthwhile asking whether critical agency towards algorithmic structures is possible without some kind of functional understanding of code and algorithms. This notion leads back to the argument that 'one reason that we must understand computers now is so that we can understand what is happening, and make informed choices, as members of a computationally-steeped democracy' (Wardrip-Fruin, 2015, p. 105) presented at the beginning of the paper. Achieving such an agentic stance requires computational technologies and their societal effects to be explored side-by-side from both functional and critical viewpoints. The framework provided

in this paper offers some conceptual and pedagogical tools to help teachers, researchers and other actors in the educational context in pursuing this goal.

CONFLICT OF INTEREST

The author does not report any conflicts of interest.

DATA AVAILABILITY STATEMENT

The article is conceptual, and no data were used.

ETHICS STATEMENT

The paper is conceptual and did not involve data collection from human subjects.

ORCID

Pekka Mertala  <https://orcid.org/0000-0002-3835-0220>

REFERENCES

- Apple, M. (1990). *Ideology and curriculum* (2nd ed.). Routledge.
- Barassi, V. (2020). *Child data citizen*. MIT Press.
- Barrett, B. (2020, March). An artist used 99 phones to fake a Google Maps traffic jam. *The Verge*. <https://www.wired.com/story/99-phones-fake-google-maps-traffic-jam/>
- Berry, D. (2014). *Critical theory and the digital*. A & C Black.
- Berry, M. (2013). *Computing in the national curriculum. A guide for primary teachers*. Computing at School and NAACE. <https://www.computingschool.org.uk/data/uploads/CASPrimaryComputing.pdf>
- Biesta, G., Priestley, M., & Robinson, S. (2015). The role of beliefs in teacher agency. *Teachers and Teaching*, 21(6), 624–640. <https://doi.org/10.1080/13540602.2015.1044325>
- Brennan, K., & Resnick, M. (2012, April). New frameworks for studying and assessing the development of computational thinking. In *Proceedings of the 2012 annual meeting of the American Educational Research Association*. Vancouver, Canada. <http://scratched.gse.harvard.edu/ct/files/AERA2012.pdf>
- Bresnihan, N., Millwood, R., Oldham, E., Strong, G., & Wilson, D. (2015). A critique of the current trend to implement computing in schools. *Pedagogika*, 65(3), 292–300.
- Campbell, C., & Walsh, C. (2017). Introducing the 'new' digital literacy of coding in the early years. *Practical Literacy: The Early and Primary Years*, 22(3), 10–12.
- Chen, M. D., Kuo, C. C., Pellegrini, C. A., & Hsu, M. J. (2016). Accuracy of wristband activity monitors during ambulation and activities. *Medicine & Science in Sports & Exercise*, 48(10), 1942–1949. <https://doi.org/10.1249/MSS.0000000000000984>
- Dickens, J. (2016, January). *ICT teachers struggling with transition to computing*. *Schools week*. <https://schoolweek.co.uk/teachers-lack-confidence-in-computing-science/>
- Dickey, M. (2020, September). *Twitter and Zoom's algorithmic bias issues*. <https://techcrunch.com/2020/09/21/twitter-and-zoom-algorithmic-bias-issues/>
- 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>
- Dufva, T., & Dufva, M. (2019). Grasping the future of the digital society. *Futures*, 107, 17–28. <https://doi.org/10.1016/j.futures.2018.11.001>
- Duncan, C., Bell, T., & Atlas, J. (2017, January). What do the teachers think? Introducing computational thinking in the primary school curriculum. In *Proceedings of the Nineteenth Australasian Computing Education Conference* (pp. 65–74). Association for Computing Machinery. <https://doi.org/10.1145/3013499.3013506>
- Fagerlund, J., Häkkinen, P., Vesisenaho, M., & Viiri, J. (2021). Computational thinking in programming with scratch in primary schools: A systematic review. *Computer Applications in Engineering Education*, 29(1), 12–28. <https://doi.org/10.1002/cae.22255>
- Futschek, G. (2006, November). Algorithmic thinking: the key for understanding computer science. In R. Mittmeier (Ed.), *International conference on informatics in secondary schools-evolution and perspectives* (pp. 159–168). Springer. https://doi.org/10.1007/11915355_15
- Geldreich, K., Talbot, M., & Hubwieser, P. (2018, October). Off to new shores: preparing primary school teachers for teaching algorithmics and programming. In *Proceedings of the 13th workshop in primary and secondary computing education* (pp. 1–6). Association for Computing Machinery. <https://doi.org/10.1145/3265757.3265783>

- Gillinov, S., Etiwy, M., Wang, R., Blackburn, G., Phelan, D., Gillinov, A. M., Houghtaling, P., Javadikasgari, H., & Desai, M. Y. (2017). Variable accuracy of wearable heart rate monitors during aerobic exercise. *Medicine & Science in Sports & Exercise*, 49(8), 1697–1703. <https://doi.org/10.1249/mss.0000000000001284>
- Gordon, M. (2009). The misuses and effective uses of constructivist teaching. *Teachers and Teaching: Theory and Practice*, 15(6), 737–746. <https://doi.org/10.1080/13540600903357058>
- Haggerty, K., & Ericson, R. (2000). The surveillant assemblage. *The British Journal of Sociology*, 51(4), 605–622. <https://doi.org/10.1080/00071310020015280>
- Hobbs, R. (2020). Propaganda in an age of algorithmic personalization: Expanding literacy research and practice. *Reading Research Quarterly*, 55(3), 521–533. <https://doi.org/10.1002/rrq.301>
- Jandrić, P., Knox, J., Besley, T., Ryberg, T., Suoranta, J., & Hayes, S. (2018). Postdigital science and education. *Educational Philosophy and Theory*, 50(10), 893–899. <https://doi.org/10.1080/0010131857.2018.1454000>
- Koivisto, K., Koski, P., & Matarma, T. (2017). Fyysisen aktiivisuuden ja kouluviihtyvyyden muutokset actiontrack-sovelluksen opetuskäytön myötä–pilottitutkimus. *Liikunta ja Tiede*, 54(2–3), 91–98.
- Larke, L. R. (2019). Agentic neglect: Teachers as gatekeepers of England's national computing curriculum. *British Journal of Educational Technology*, 50(3), 1137–1150. <https://doi.org/10.1111/bjet.12744>
- Lecher, C. (2019, April). How Amazon automatically tracks and fires warehouse workers for 'productivity'. *The Verge*. <https://www.theverge.com/2019/4/25/18516004/amazon-warehouse-fulfillment-centers-productivity-firing-terminations>
- Lee, C. H., & Soep, E. (2016). None but ourselves can free our minds: Critical computational literacy as a pedagogy of resistance. *Equity & Excellence in Education*, 49(4), 480–492. <https://doi.org/10.1080/106665684.2016.1227157>
- Lee, I. M., Shiroma, E. J., Kamada, M., Bassett, D. R., Matthews, C. E., & Buring, J. E. (2019). Association of step volume and intensity with all-cause mortality in older women. *JAMA Internal Medicine*, 179(8), 1105–1112. <https://doi.org/10.1001/jamainternmed.2019.0899>
- Lupton, D. (2021). 'Honestly no, I've never looked at it': teachers' understandings and practices related to students' personal data in digitised health and physical education. *Learning, Media and Technology*, 1–13. <https://doi.org/10.1080/17439884.2021.1896541>
- Manches, A., & Plowman, L. (2017). Computing education in children's early years: A call for debate. *British Journal of Educational Technology*, 48(1), 191–201. <https://doi.org/10.1111/bjet.12355>
- Mason, S. L., & Rich, P. J. (2019). Preparing elementary school teachers to teach computing, coding, and computational thinking. *Contemporary Issues in Technology and Teacher Education*, 19(4), 790–824.
- McDonlad, C. (2017, October). Almost 70% of teachers think they do not have the skills to teach coding. *Computer Weekly*. <https://www.computerweekly.com/news/450429016/Almost-70-of-teachers-think-they-do-not-have-the-skills-to-teach-coding>
- McGarr, O., & Johnston, K. (2020). Curricular responses to Computer Science provision in schools: current provision and alternative possibilities. *The Curriculum Journal*, 31(4), 745–756. <https://doi.org/10.1002/curj.40>
- Mee, A. (2020). *Computing in the school curriculum: A survey of 100 teachers*. https://www.researchgate.net/publication/339536481_Computing_in_the_school_curriculum_a_survey_of_100_teachers
- Merikivi, J., Myllyniemi, S., & Salasuo, M. (2016). *Media hanskassa. Lasten ja nuorten vapaa-aikatutkimus 2016 mediasta ja liikunnasta*. Nuorisotutkimusseura.
- Mertala, P. (2019a). Young children's conceptions of computers, code, and the Internet. *International Journal of Child-Computer Interaction*, 19, 56–66. <https://doi.org/10.1016/j.ijcci.2018.11.003>
- Mertala, P. (2019b). Teachers' beliefs about technology integration in early childhood education: A meta-ethnographical synthesis of qualitative research. *Computers in Human Behavior*, 101, 334–349. <https://doi.org/10.1016/j.chb.2019.08.003>
- Mertala, P., Palsa, L., & Dufva, T. S. (2020). Monilukutaito koodin purkajana: Ehdotus laaja-alaiseksi ohjelmoinnin pedagogiikaksi. *Media & Viestintä*, 43(1), 21–46. <https://doi.org/10.23983/mv.91079>
- Mikkola, H., Kumpulainen, K., & Rahikkala, A. (2011). Pedometer intervention among 4th graders—Pedometers as pedagogical practise to enhance children's wellbeing in formal and informal contexts. In *Proceedings of Society for Information Technology & Teacher Education International Conference 2011* (pp. 2586–2593). AACE.
- O'Neil, C. (2016). *Weapons of math destruction: How big data increases inequality and threatens democracy*. Penguin.
- Osborne Clarke. (2020, July). *Employee monitoring during Covid-19, the new normal?*. <https://www.osborneclarke.com/insights/employee-monitoring-covid-19-new-normal/>
- Otterborn, A., Schönborn, K. J., & Hultén, M. (2020). Investigating preschool educators' implementation of computer programming in their teaching practice. *Early Childhood Education Journal*, 48, 253–262. <https://doi.org/10.1007/s10643-019-00976-y>
- Pangrazio, L., & Sefton-Green, J. (2021). Digital rights, digital citizenship and digital literacy: What's the difference? *Journal of New Approaches in Educational Research*, 10(1), 15–27. <https://doi.org/10.7821/naer.2021.1.616>

- Pangrazio, L., & Selwyn, N. (2019). 'Personal data literacies': A critical literacies approach to enhancing understandings of personal digital data. *New Media & Society*, 21(2), 419–437. <https://doi.org/10.1177/1461444818799523>
- Pangrazio, L., & Selwyn, N. (2020). Towards a school-based 'critical data education'. *Pedagogy, Culture & Society*, 1–18. <https://doi.org/10.1080/14681366.2020.1747527>
- Papadakis, S., & Kalogiannakis, M. Learning computational thinking development in young children with Bee-Bot educational robotics. In M. Kalogiannakis & S. Papadakis (Eds.), *Handbook of research on tools for teaching computational thinking in P-12 education* (pp. 285–305). IGI Global. <https://doi.org/10.4018/978-1-7998-4576-8.ch011>
- Papert, S. (1980). *Mindstorms: Computers, children, and powerful ideas*. Basic Books.
- Parak, J. (2018). *Evaluation of wearable optical heart rate monitoring sensors* (PhD dissertation). University of Tampere, Finland.
- Pasadyn, S. R., Soudan, M., Gillinov, M., Houghtaling, P., Phelan, D., Gillinov, N., Bittel, B., & Desai, M. Y. (2019). Accuracy of commercially available heart rate monitors in athletes: A prospective study. *Cardiovascular Diagnosis and Therapy*, 9(4), 379–385. <https://doi.org/10.21037/cdt.2019.06.05>
- Polar. (n.d.). *Polar Active käyttöohje*. https://support.polar.com/e_manuals/Active/Polar_Active_user_manual_Suomi/Polar_Active_user_manual_Suomi/Polar_Active_kayttoohje_FI.pdf
- Rantala, J. (2018). Koodin ja ohjelmiston filosofiasta. *Niin & Näin*, 2018(1), 101–108.
- Rich, P. J., Browning, S. F., Perkins, M., Shoop, T., Yoshikawa, E., & Belikov, O. M. (2019). Coding in K-8: International trends in teaching elementary/primary computing. (*TechTrends*, 63(3), 311–329. <https://doi.org/10.1007/s11528-018-0295-4>
- Rich, P. J., Mason, S. L., & O'Leary, J. (2021). Measuring the effect of continuous professional development on elementary teacher' self-Efficacy to teach coding and computational thinking. *Computers & Education*, 168, 104196. <https://doi.org/10.1016/j.compedu.2021.104196>
- Sáez-López, J. M., Román-González, M., & Vázquez-Cano, E. (2016). Visual programming languages integrated across the curriculum in elementary school: A two year case study using "Scratch" in five schools. *Computers & Education*, 97, 129–141. <https://doi.org/10.1016/j.compedu.2016.03.003>
- Selwyn, N. (2019). *Should robots replace teachers? AI and the future of education*. Polity.
- Sentance, S., & Csizmadia, A. (2017). Computing in the curriculum: Challenges and strategies from a teacher's perspective. *Education and Information Technologies*, 22(2), 469–495. <https://doi.org/10.1007/s10639-016-9482-0>
- Sharon, T., & Zandbergen, D. (2017). From data fetishism to quantifying selves: Self-tracking practices and the other values of data. *New Media & Society*, 19(1), 1695–1709. <https://doi.org/10.1177/1461444816636090>
- Solomon, C., Harvey, B., Kahn, K., Lieberman, H., Miller, M., Minsky, M., Papert, A., & Silverman, B. (2020). History of Logo. *Proceedings of the ACM on Programming Languages*, 4(79), 1–66. <https://doi.org/10.1145/3386329>
- Tanhua-Piironen, E., Kaarakainen, S., Kaarakainen, M., & Viteli, J. (2020). *Digiajan peruskoulu II* (No. 17). Opetus- ja kulttuuriministeriön julkaisuja.
- The Beauty and Joy of Computing. (n.d.). *The BJC curriculum*. <https://bjc.berkeley.edu/curriculum/>
- The New London Group. (1996). A pedagogy of multiliteracies: Designing social futures. *Harvard Educational Review*, 66(1), 60–93. <https://doi.org/10.17763/haer.66.1.17370h67v22j160u>
- Tudor-Locke, C., Aguiar, E. J., Han, H., Ducharme, S. W., Schuna, J. M., Barreira, T. V., & Chipkin, S. R. (2019). Walking cadence (steps/min) and intensity in 21–40 year olds: CADENCE-adults. *International Journal of Behavioral Nutrition and Physical Activity*, 16(1), 8. <https://doi.org/10.1186/s12966-019-0769-6>
- Tuomi, P., Multisilta, J., Saarikoski, P., & Suominen, J. (2018). Coding skills as a success factor for a society. *Education and Information Technologies*, 23(1), 419–434. <https://doi.org/10.1007/s10639-017-9611-4>
- Valtonen, T., Tedre, M., Mäkitalo, K., & Vartiainen, H. (2019). Media literacy education in the age of machine learning. *Journal of Media Literacy Education*, 11(2), 20–36. <https://doi.org/10.23860/jmle-2019-11-2-2>
- van der Kruk, E., & Reijne, M. M. (2018). Accuracy of human motion capture systems for sport applications; state-of-the-art review. *European Journal of Sport Science*, 18(6), 806–819. <https://doi.org/10.1080/17461391.2018.1463397>
- Vandelanotte, C., Van Itallie, A., Brown, W., Mummery, W. K., & Duncan, M. J. (2020). Every step counts: Understanding the success of implementing the 10,000 Steps project. *Studies in Health Technology and Informatics*, 2(268), 15–30. <https://doi.org/10.3233/SHTI200003>
- Veel, A. (2013). Understanding computer programming as a literacy. *Literacy in Composition Studies*, 1(2), 42–64. <https://doi.org/10.21623/1.1.2.4>
- Veel, A. (2017). *Coding literacy: How computer programming is changing writing*. MIT Press.
- Vega, J., & Cañas, J. M. (2019). PyBoKids: An innovative python-based educational framework using real and simulated arduino robots. *Electronics*, 8(8), 899. <https://doi.org/10.3390/electronics8080899>
- Wardrip-Fruin, N. (2015). We can and must understand computers NOW. In D. Dechow & D. Struppa (Eds.), *Intertwined, history of computing* (pp. 105–112). Springer.

- Williamson, B. (2015). Algorithmic skin: Health-tracking technologies, personal analytics and the biopedagogies of digitized health and physical education. *Sport, Education and Society*, 20(1), 133–151. <https://doi.org/10.1080/13573322.2014.962494>
- Williamson, B. (2016). Political computational thinking: Policy networks, digital governance and 'learning to code'. *Critical Policy Studies*, 10(1), 39–58. <https://doi.org/10.1080/19460171.2015.1052003>
- Williamson, B. (2017). *Big data in education: The digital future of learning, policy and practice*. Sage.
- Williamson, B., Bergviken Rensfeldt, A., Player-Koro, C., & Selwyn, N. (2019). Education recoded: Policy mobilities in the international 'learning to code' agenda. *Journal of Education Policy*, 34(5), 705–725. <https://doi.org/10.1080/02680939.2018.1476735>
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33–35. <https://doi.org/10.1145/1118178.1118215>
- Wu, L., Looi, C. K., Multisilta, J., How, M. L., Choi, H., Hsu, T. C., & Tuomi, P. (2020). Teacher's perceptions and readiness to teach coding skills: A comparative study between Finland, mainland China, Singapore, Taiwan, and South Korea. *The Asia-Pacific Education Researcher*, 29(1), 21–34. <https://doi.org/10.1007/s40299-019-00485-x>

How to cite this article: Mertala, P. (2021). The pedagogy of multiliteracies as a code breaker: A suggestion for a transversal approach to computing education in basic education. *British Journal of Educational Technology*, 00, 1–15. <https://doi.org/10.1111/bjet.13125>