Survey: Artificial intelligence, computational thinking and learning

Dohn, Nina Bonderup; Kafai, Yasmin; Mørch, Anders; Ragni, Marco

Published in:
KI - Künstliche Intelligenz

DOI:
10.1007/s13218-021-00751-5

Publication date:
2022

Document version:
Accepted manuscript

Citation for pulished version (APA):
Dohn, N. B., Kafai, Y., Mørch, A., & Ragni, M. (2022). Survey: Artificial intelligence, computational thinking and learning. *KI - Künstliche Intelligenz*, *36*(1), 5-16. https://doi.org/10.1007/s13218-021-00751-5

Title page

Survey: Artificial Intelligence, Computational Thinking and Learning

Authors

- Nina Bonderup Dohn, Department of Design and Communication, University of Southern Denmark, Kolding, Denmark, nina@sdu.dk, https://orcid.org/0000-0002-5640-7446
- Yasmin Kafai, Graduate School of Education, University of Pennsylvania, Philadelphia, Pennsylvania, USA, kafai@upenn.edu, https://orcid.org/0000-0003-4018-0491
- Anders Mørch, Department of Education, University of Oslo, Oslo, Norway, andersm@uio.no, https://orcid.org/0000-0002-1470-5234
- Marco Ragni, Institute of Psychology, Technical University Chemnitz, Chemnitz, Germany

Abstract

Learning is central to both artificial intelligence and human intelligence, the former focused on understanding how machines learn, the latter concerned with how humans learn. With the growing relevance of computational thinking, these two efforts have become more closely connected. This survey examines these connections and points to the need for educating the general public to understand the challenges which the increasing integration of AI in human lives pose. We describe three different framings of computational thinking: cognitive, situated, and critical. Each framing offers valuable, but different insights into what computational thinking can and should be. The differences between the three framings also concern the views of learning that they embody. We combine the three framings into one framework which emphasizes that 1) computational thinking activities involve engagement with algorithmic processes, and 2) the mere use of a digital artifact for an activity is not sufficient to count as computational thinking. We further present a set of approaches to learning computational thinking. We argue for the significance of computational thinking as regards artificial intelligence on three counts: (i) Human developers use computational thinking to create and develop artificial intelligence systems, (ii) understanding how humans learn can enrich artificial intelligence systems, and (iii) such enriched systems will be explainable. We conclude with an introduction of the articles included in the Special Issue, focusing on how they call upon and develop the themes of this survey.

Survey: AI, Computational Thinking and Learning

1. The significance of learning computational thinking for the field of AI

Learning is central to both artificial intelligence (AI) and human intelligence, the former focused on understanding how machines learn, the latter concerned with how humans learn. With the growing relevance of computational thinking (CT) these two efforts have become more closely connected. This survey examines these connections and points to the need for educating the general public to understand the challenges that the increasing integration of AI in human lives pose.

From the beginnings of AI, the connection to CT was strong, with representatives of classical strong AI believing that minds and computer programs functioned alike as rule-following information processing agents. On this view, writing AI programs was, in effect, doing theoretical (cognitive) psychology or experimental epistemology [e.g. 1, 2, 3]. While there are several definitions of AI, probably the most prominent one considers it on two dimensions. Firstly, the dimension from thinking to action, and secondly, the dimension from human to rational agent [4]. An artificial intelligence is thus a system that can *act and/or think*, either as *a human or* as a *rational agent* that maximizes its outcome. To develop such a system, an important question is whether a normative framework exists that can determine what 'correct' rational thinking/acting is or, alternatively, whether a descriptive framework can be developed that can capture how humans think/act. For both kinds of framework, it is necessary to consider that the architectures of AI systems do in fact differ from the architecture of human cognition: The first are still built on the theoretical framework of Turing machines. The second one needs to be reverse engineered to be modelled in an artificial system and is currently approximated, but not yet adequately represented, by a large variety of cognitive architecture models. To make progress in either direction requires to evaluate the similarity and difference between artificial and human cognition. This necessitates establishing a common ground on which they can both be measured and thus compared. One way to do this is to evaluate how well systems perform on typical human tasks. Another way is to evaluate how humans perform on problems within the field of computer systems, that is on problems that require an algorithmic solution. This is the domain of computational thinking.

The term 'computational thinking' itself was coined by Seymour Papert in his ground-breaking work on how children through programming can come to understand powerful ideas in mathematics and science [5]. Papert used the term to denote the procedural thinking employed in writing programs, emphasizing elsewhere the close connection to the AI of the time with the claim that AI can be viewed as a procedural theory of knowledge [6]. Originally, the connection was specialized to a few research communities such as theoretical computer science, robotics and cognitive science. Today there is renewed and broader interest in the potentially fruitful relationship between computational thinking (by humans) and machine learning (by computers) and how they can inform each other. A recent article claims quite starkly that "[b]asically, the latter [AI skills] is a subset of the former [CT skills] – computational thinking skills refer to the ability to reason with any algorithm and AI is a specific family of algorithms." [7].

Exploring the relationship between human and machine thinking is a key area for this Special Issue. Initially, CT was a concern primarily within computer science, but Jeanette Wing's [8] influential essay broadened the focus well beyond this discipline. She argued that every child should learn to "think… like a computer scientist" (p. 34) because CT "represents a universally applicable attitude and skill set for everyone" (p. 33). Her much-cited definition of CT came in a follow-up article: "Computational Thinking is the thought processes involved in formulating problems and their solutions so that the solutions are represented in a form that can be effectively carried out by an information-processing agent... human or machine, or… a combination of humans and machines" [9]. The connection to AI is implicitly present in this juxtaposition of humans and machines as information-processing agents.

However, Wing's definition focuses mainly on problems and their procedural formulation and solution leaving out social and political dimensions. We prefer a broader definition proposed by Dohn [10, our translation], which targets development of digital artifacts (including AI) more generally: "Computational thinking consists in the cognitive processes involved in the development of digital artifacts and programs to live in today's world." This definition allows for wider or narrower understandings of cognition, corresponding to the different approaches to CT that we discuss below. Furthermore, since the definition is ostensive with no claim to exclusiveness, it is an open empirical question whether the same cognitive processes are involved also in other activities. Finally, the definition explicitly integrates reference to the context for which digital artifacts are developed ('living in today's world') and thus emphasizes the concern for sustainable integration of digital artifacts and programs. This in turn highlights critical, creative, and reflective cognitive processes as integral to CT.

Moreover, this expanded view of CT allows us to examine critical challenges in its connection to AI. For one, we need to realize that it is (still) the human and not an automated system that develops AI systems. Second, AI scientists of tomorrow are learners of today. Third, humans in general, not just computer scientists, have to deal with AI in everyday interactions with digital artifacts in one form or the other. Doing so competently and potentially synergistically necessitates a basic understanding of CT. Fourth, humans need a general understanding of how AI works, if we as a society are to integrate AI sustainably (i.e., balance economic, societal, environmental and ethical considerations) [11], including having enlightened discussions of ethical implications and living with decisions made by AI systems. Finally, understanding how humans think computationally can provide new explainable models [12] of machine learning and thus push AI forward.

In the next section, we describe three different framings of CT. In section 3, we combine these framings into one framework. This allows us in section 4 to present a set of approaches to learning CT. In section 5, we return to the question of how (learning) AI relates to (learning) CT. We conclude with an overview of the articles included in the Special Issue, focusing on how each article picks up on and develop the themes discussed in this survey.

2. Different framings of computational thinking

Computational thinking is influencing the introduction of the subject of AI into schools and the broader society around the globe [7]. Taking stock on how computational thinking has been framed since Wing's [8] popularization of the term, it is therefore necessary to elucidate the perspectives on AI which different framings allow. Various framings of CT have been discussed in the research literature. Most prominent are cognitive perspectives [13] but these are complemented by literacy perspectives [14] along with situated and critical perspectives. Situated perspectives emphasize computational participation [15] and computational making [16], and critical perspectives highlight computational action [17]. Each of these latter framings emphasize agency and creativity in the learner's engagement with computation and computational artifacts.

In its original definition, CT was mostly framed in cognitive terms, as the connotation of 'thinking' in relation to 'computation' illustrates. The cognitive framing harks back to Wing's [8] early description of CT as "taking an approach to solving problems, designing systems and understanding human behaviour that draws on concepts fundamental to computing" (p. 33). This description builds on the prominent cognitive research traditions mentioned in Section 1 that see artificial intelligence as a form of complex problem solving, first studied in humans and later modelled as information processing and realized in algorithms that could perform similar feats. Here, Herbert Simon's reflections on the sciences of the artificial were instrumental in marking the connections between human and machine cognition [18]. CT from a cognitive perspective [13] presents an individualistic framing of human and machine learning that downplays the social and cultural contexts in which it operates.

A different framing of CT is computational participation [15, 19] highlighting learning's situated nature. Such a framing moves from an individualistic perspective to one of learning as a social activity driven by personal interests and community interactions. A paradigmatic example is the online community Scratch, where children can engage in coding interactive media by utilizing a programming language of the same name. Here, children learn coding through self-directed choice, by sharing projects, and building on, remixing and extending others' projects [20; https://scratch.mit.edu/]. Though Wing's [8] definition connects the design of systems to human behaviour, it falls short of seeing the larger social and cultural contexts in which these systems are conceived and realized. In contrast, framings of CT as participation emphasize that the design of any system is situated within a social context that provides interaction, feedback on interactions and performances. Such framings draw on Hutchins' seminal work on cognition in the wild [21] that examined how culture shaped individual cognition and computational systems and saw them as part of a larger social context. Another significant source of inspiration for understanding the interlocking of computation with social context is the arguments of diSessa that computation potentially provides a new literacy – computational literacy – which will make new ways of thinking and learning possible, in the same way that textual literacy revolutionized human thinking and learning, transcending a predominantly oral society [14, 22].

Finally, critical framings expand CT as a potential channel for engaging with the political, moral and ethical challenges connected to the design and use of intelligent systems [23]. These framings draw on recent criticisms that CT does not go far enough in 'pulling back the curtain' of the technological mechanisms underlying our existing computational systems in order to understand how these may cause inequities in and of itself. This includes, for instance, confronting forces such as filter bubbles [24] and the algorithmic bias present in many search algorithms, face-recognition software, and crime-prediction software that police use, to name but a few examples [25-27].

Each framing offers valuable, but different insights into what CT can and should be about and how this can apply to the design, implementation and critique of intelligent systems. In fact, the differences that emerge between these three framings of CT are not just in purpose but also in the views of learning that they embody and the attention they have received so far. The cognitive framing by far outpaces the other theoretical framings. One possible reason for the dominance is that when the first wave of research and design of AI started in the 1980s, cognitive theory was the dominant framework for conceptualizing thinking and problem solving across different academic disciplines and for developing intelligent systems. Proponents of CT like Wing followed suit, most likely finding resonance with the cognitive perspectives featuring the individual mind as an information processing unit. However, critics of these cognitive views have highlighted some of its weaknesses, namely that learning is not just an individual enterprise but situated in social situations and practice and often the interplay of social interactions mediated by artifacts is decisive for the different learning opportunities actually (not just potentially) opened to different learners [28-31]. Finally, critical views of learning and computation highlight the need to pay attention to cultural issues, emphasizing that what is learned and how it is learned and valued reflects the particular norms, values, and power structures of a society. They point out how this is reflected in designed systems [23, 32, 33]. Despite these criticisms, the cognitive framing's focus on the individual learner as the unit of analysis arguably stays relevant also for the other framings.

3. Delimitation of computational thinking activities

We propose a framework for CT activities which encompasses the three framings, whilst highlighting their divergencies. We discuss three key questions:

A. Do CT activities (primarily) concern *construction* of or *critical reflection* on algorithmic artifacts?
B. Do CT activities necessarily involve digital artifacts or can they take place solely with analogue means (also called 'unplugged')?
C. Do CT activities necessarily involve coding programs?

The framings presented in the last section provide different answers to these questions which we elaborate on in the following.

On the cognitive framing of CT as problem solving, the answer to question A is construction – and the algorithmic artifact to be constructed is a solution to the problem at hand. However, this solution does not have to be implemented in a program. Therefore, the answer to question B is that CT activities need not involve digital artifacts because the algorithmic solution can be found and represented with analogue means (e.g., in the learner's mind or with pen and paper) which essentially will involve the same cognitive processes as the ones necessary for designing a programmed solution [34, 35]. In consequence, Question C will be answered in the negative.

The situated framing of CT focuses on participation in computational construction practices. Therefore, the answer to question A is also construction – but here, the algorithmic artifact to be constructed is a digital artifact. 'Digital artifact' is taken in a broad sense as referring to any artifact involving some coded algorithmic processes. Question C will be answered in the affirmative with the qualification that coding is understood as 'engaging with code', allowing the term to apply also to users who for instance construct an electronic textile (e.g. a piece of clothing with LED lights that are programmed with light effects) and utilize unaltered code made by others [36]. The answer to question B is that CT activities necessarily involve digital artifacts.

On the critical framing of CT, question A is answered with critical reflection. The answer to question B is that reflection itself can – and most often will – take place solely with analogue means, but the focus of critique is digital artifacts. Question C will therefore be answered negatively. Still, arguably it is necessary that users have some understanding of the workings of the artifact's code and algorithms to adequately engage in critical assessment [37].

This last point highlights a common trait cutting across the three framings: CT activities necessarily involve engagement with algorithmic processes. The framings do differ on what exactly this engagement condition entails. For instance, informed *use* of existing programs for data visualization (e.g., of the dynamics of economics or of the evolution over time of a biological ecosystem) will count as engagement on the cognitive framing, but not on the two others.

The framings all agree on a further, core point: that the mere use of a digital artifact for an activity is not sufficient to count as CT. Thus, for instance creating a PowerPoint presentation, watching a streaming service, posting on social media, or participating in a discussion in an online forum will not be CT activities, as they do not require an interaction on the algorithmic level. Neither will non-reflective use of AI systems where system output is black-boxed, taken for granted and acted upon without any understanding of the algorithms involved. In this way, the engagement condition serves to delimit CT activities from other forms of digital activities. This is of significance, partly for clarification, as the terms 'participation', 'creativity', 'critical', 'reflective' are also often used in discussion of so-called 'digital literacy' [38] and of contemporary 'participatory culture' [39]. Here, focus is on learners' critically reflective creation, use and evaluation of online communication, information resources, multimedia productions etc. Such activities of course utilize software, but most often in a black-boxed form where learners do not engage with the underlying algorithms.

Figure 1 presents our framework for CT activities. It draws two basic distinctions: between a) activities solely with analogue means and b) digital activities (corresponding to question B) and between i) coding programs and ii) using programs (for digital activities, corresponding to positive and negative answers to question C). The cognitive framing is depicted by the solid line figure (orange), representing that problem solving can take place with analogue means, via informed use of programs for data processing and complex system visualization, or through coding a solution. The situated framing is shown as the solid line circle (brown) spanning the distinction between using and coding programs, to highlight that engaging with code can consist in the informed overtaking of others' code in the creation of one's own artifact. Digital activities are indicated with a dashed line (green). There is a large overlap with the solid line figure depicting the cognitive framing, as all problem solving involving the use or coding of programs will be digital activities. Finally, the critical framing is portrayed in the brace, spanning both the solid and dashed line figures, to indicate that critical reflection should concern not only the algorithms involved in CT activities but also the algorithms underlying pure use of digital artifacts. The size of the squares and circles are not indicative of the prevalence or relative importance of each area.

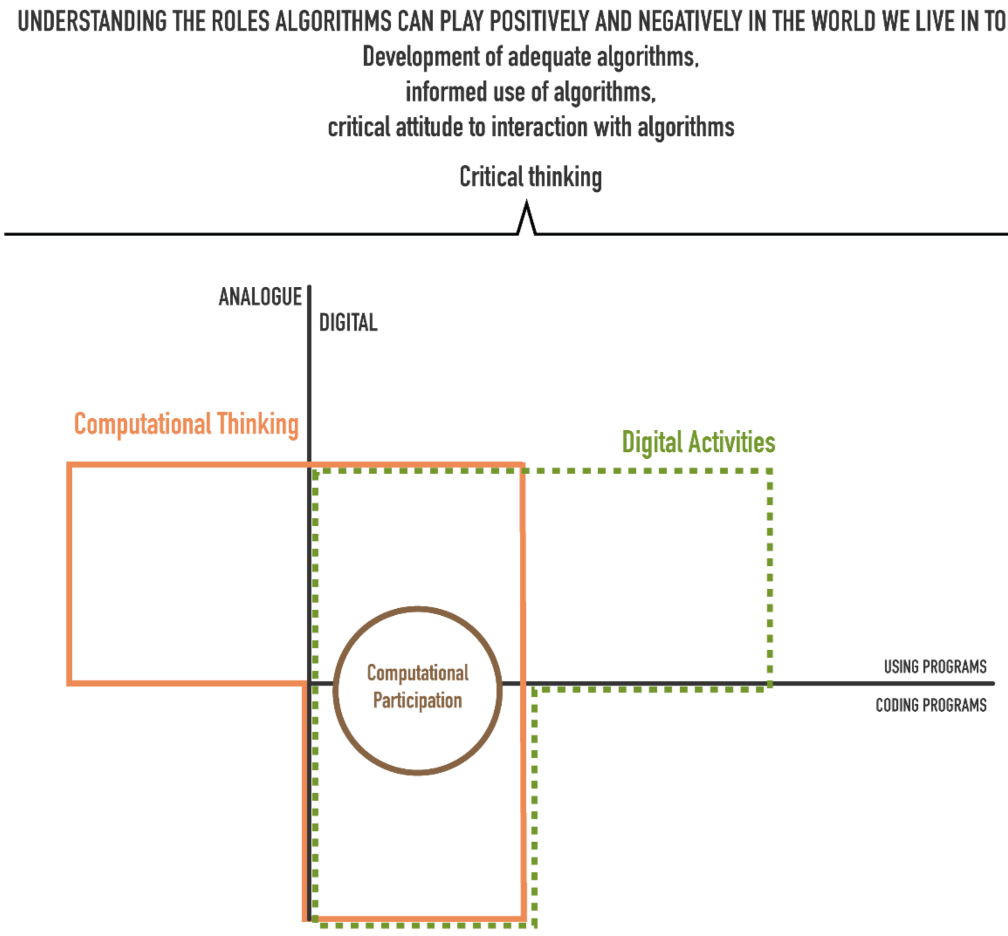

Figure 1. Framework for CT activities encompassing the three framings mentioned in Section 2: cognitive, situated, and critical. The framework also highlights that not all digital activities involve CT.

4. Practical, pedagogical approaches and applications for learning computational thinking

In this section, we present an overview of different pedagogical approaches to designing activities to support learners in CT. Each approach draws primarily on one or other of the three framings. In addition, they build on learning theoretical and pedagogical literature. In this sense, they represent different answers to the question of developing a 'Fachdidaktik' (a term not easily translated; the closest would be: pedagogical content knowledge [40]) for CT.

Approach 1: Incremental development of problem-solving

This approach draws on the cognitive framing and views CT as, fundamentally, a problem-solving technique consisting of a set of consecutive steps which can be performed with or without the use of digital artifacts and coding. The purpose of students' learning CT is not their engagement with computer science per se, but the acquisition of a technique that is useful in other (potentially all) disciplines. Accordingly, the approach emphasizes ‘easy beginnings’ with analogue means and anchorage in actual disciplinary problems [34, 35, 41] such as modelling economic relationships or constructing realistic alternative storylines for a book. This allows learners to focus on internalizing the problem-solving technique and developing the corresponding cognitive skills without the risk of being offset by having to learn a programming language. Moreover, they get to experience the value and relevance of CT. From the easy beginnings, learners can progressively and incrementally be introduced to various digital tools for data analysis and visualization, which permits their experiencing the enhanced possibilities for problem solving provided by such tools. This opens for the last phase where learners are encouraged themselves to code artifacts to support their problem solving, potentially again with a progression from high-level scripting languages to more advanced forms of coding. In effect, the approach recommends a progression of CT activities starting in the upper left corner of Figure 1 and moving through the solid line figure, from left to right, and then downwards.

Approach 2: Iterative problem-solving with information technology

This approach is similar to the previous one in drawing on the cognitive framing of CT and emphasizing a progression from use of digital tools to creation of one’s own. However, in contrast to the previous approach, the approach of iterative problem solving with information technology (IT) has a narrower focus on computer science and the use of computer science methods within other disciplines, primarily STEM (Science, Technology, Engineering and Mathematics). Correspondingly, the rationale of CT activities is learners’ development of programming skills, not of a general problem-solving technique. Pedagogically, the approach recommends scaffolding the learning of such skills through a three-stage progression model of use-modify-create [42]. Each stage itself involves a movement from simpler to more complex activities. The first stage concerns the use of existing programs, notably with the requirement that learners have access to inspecting the underlying code. Progression within this stage will be from the use and inspection of simpler programs to more complex ones. The next stage concerns modification of existing code, progressing from simple adjustments to the writing and implementing of new pieces of code in given programs. The last stage concerns creation of new computational projects, progressing from simple to more complex ones. In Figure 1, the approach is concerned only with the digital part of the solid line figure, with the clear aim of getting learners engaged in the lower square by pedagogically scaffolding the movement from the upper one.

Approach 3: Contextual problem-solving

The former two approaches focus on CT as a set of skills primarily performed by individuals [13]. Another direction emphasizes how students in collaboration can develop CT through designing and programming shareable digital artifacts. This direction draws from constructionist learning theory [5], which emphasizes interest-driven and peer-supported activities and thus sees CT as a vehicle for personal expression and participation [15]. It draws on the situated framing of CT as computational participation and therefore focuses on the solid line circle in Figure 1. It encompasses a set of pedagogical takes which converge in emphasizing the following characteristics of learning processes as decisive in supporting the learning of CT: creativity, playfulness, learners’ construction of digital artifacts, self-chosen projects, participation in construction practices, collaboration with others, and curation of learners’ passion for their projects – what Resnick has termed the 4 p’s of constructionism: Projects, peers, passion, and play [43]. As compared to approach 2, the strategy of use-modify-create may well be what learners end up following in practice, but this is not suggested as a pedagogical principle. The pedagogical principles are the 4 p’s which allow learners to define their own learning trajectories – and set their own learning goals – rather than stipulate an overall process, such as use-modify-create.

When it comes to designing educational activities, the pedagogical takes within the third approach differ somewhat on which of the said characteristics are weighted as essential and which are only viewed as important supports. We shall exemplify the overall approach as well as different instances with the pedagogical takes of *(a) creative tinkering*, *(b) designing for play*, and *(c) critical participation*. Here, (a) weights creativity, construction, and self-chosen projects as essential; (b) highlights playfulness and construction; and (c) emphasizes self-chosen projects, participation in construction practices, and collaboration with others. Further, (c) connects directly to the critical framing of CT (the brace in Figure 1), whilst (a) and (b) hold the potential for drawing on this framing, too.

Application 1: Creative tinkering

The pedagogical take of creative tinkering has become widespread in so-called makerspaces. A makerspace is often organized as a community-operated not-for-profit workspace for constructing, artistically creating, repairing, changing, repurposing – essentially hacking – artifacts. Most often, these artifacts involve computational elements and thus are digital artifacts (by the present survey's understanding), but examples of makerspaces that only involve physical arts and crafts exist. Makerspaces serve as an interesting environment for learning CT, because of their implicit valuing of creatively trying out what works and what doesn't (i.e., creative tinkering) in the design, modification and use of digital artifacts. Makerspaces and creative tinkering with technology involving coding draws extensively on the work of Mike Eisenberg, Mitchel Resnick, and colleagues [44-46].

Makerspaces as organized activity started when people with a common interest in tinkering and repairing gadgets gathered to share ideas and help each other. The concept was launched in the American Make magazine in 2005. In a world of increased consumption of finished products, reuse of and repair of existing products make sense as it helps sustain the natural environment [47]. Makerspaces as learning environments refer to the sites (informal and formal) and activities where people meet to explore ideas, learn technical skills, and create new products together [48], including museums, libraries, after school clubs, and public schools [49]. The utilization of makerspaces in schools is challenged by inflexible curricula and teachers' lack of IT skills [50], but an increasing number of case studies show that makerspaces can provide a physical (concrete) space for learning STEM topics through creative tinkering [51] as well as provide access to new techniques in arts and crafts [52]. This is partly due to flexible computational environments [53] where students can creatively tinker self-chosen projects with only limited programming skills. It is also due to the use of inexpensive physical and electronic materials that are programmable. Furthermore, an increased awareness of societal issues raised by conspicuous consumption and expanding automatization in human activity serves as an underlying rationale for engaging makerspaces in school, both as an avenue for constructing more sustainable solutions and as a concrete outset for critical reflection (thus potentially providing a link to the critical framing of CT; the brace in Figure 1). In terms of teaching and learning CT with makerspaces, Papert's vision of CT as a pedagogical means toward learning objectives in other disciplines than computer science (e.g., powerful ideas in mathematics and science) [5] is still relevant to guide future research. For example, in most Nordic countries, CT is not a separate subject in primary and secondary education but integrated in other subjects (e.g., math, science, arts and craft). On the other hand, many researchers focus today on getting CT into public schools as a discipline of its own [54]. Either way, creative tinkering in makerspaces would be one way to organize the learning of CT in practice.

Application 2: Designing for play

Video games and virtual worlds are software tools used for entertainment, interaction and coding (e.g., game design and modding). These tools can provide entry points to knowledge at different levels of abstraction. Researchers in virtual worlds proposed action-breakdown-repair (ABR) as a pedagogical model for connecting Minecraft gameplay with subject knowledge [55]. The idea is that students will build in Minecraft (action), until something happens that they do not expect (breakdown), which necessitates that they reconsider their implicit views about the world – the physical one and/or the virtual one, as the case may be (repair). Minecraft users can modify the game experience by using a block-based (also termed visual) programming language, similar to Scratch, called MakeCode (https://makecode.microbit.org/). This can again lead the user to concrete or abstract knowledge: 1) connecting the code with scientific hypotheses underlying the code, for instance 'what happens in our world if trees do not exist' (thus utilizing CT as a means towards learning objectives in science), and 2) CT patterns associated with game modifications (thus learning CT as a discipline of its own). The latter is referred to as scalable game design and uses algorithms like latent semantic analysis (LSA) for pattern matching [56]. CT arguably involves a combination of different dimensions for the learner as designer, including concepts, practices and perspectives [53]. When the concepts are represented in a computational design environment like the *Simulation creation toolkit* [54], new arenas for teaching CT (i.e., for developing CT practices) open up. New empirical questions can be raised about students' CT practices and learning of CT when the design environments provide automated scaffolding utilizing intelligent systems like recommender systems: Such a system can for instance inform the learners about what CT concepts they successfully demonstrate or alternatively fail to achieve when creating or modifying a game, for example following the approach of critiquing systems in domain-oriented design environments [57].

Application 3: Critical participation

This pedagogical take has participation in social construction practices as its fundamental pedagogical principle. Learning key computational concepts and practices are thus situated within acts of designing complex applications and these are shared on social networks such as the Scratch online community (cf. above). Studies have been conducted to understand the progression of computational participation for different types of users, and to develop digital tools on the site that allow users to visualize their own participation online, whether through replication or creative remixing [58]. More recently, special 'community blocks' have been included in Scratch. These community blocks aim at making transparent the collection, calculation, and dissemination of participation data common in many massive online communities [59]. A study by Hautea and colleagues [60] shows that Scratch users became more cognizant of numerous issues surrounding big data today by creating their own programs with these blocks. These issues included realization of the privacy implication of data collection and retention, possible avenues for exclusion through data-driven algorithms, and possible biases and assumptions hidden within supposedly neutral data claims. While Scratch users only expressed these perspectives on the closed system of Scratch, one can see how an understanding of these ideas can help promote a larger understanding of our wider digital environment today, where algorithms and data structures are rarely shared. Here, learning computational concepts is not just an instrumental goal but pushes learners towards considering the larger socio-political implications of data collection, analysis and use, thereby connecting to the critical framing of CT (the brace in Figure 1).

5. AI perspectives on computational thinking

In the sections above we have highlighted several ways in which AI intersects with CT, as viewed from the perspective of CT:
(a) AI as a specific subset of CT skills (i.e., human reasoning with AI algorithms, [7]);
(b) AI as the domain of a profession which employs CT (i.e., a potential future profession for CT learners);
(c) AI as intelligent system support in the process of learning CT (e.g., recommender systems);
(d) AI as a domain for critical reflection as regards the political, moral and ethical consequences of extensive implementation of AI in human practices.

It is time now to turn the tables and discuss the relevance of CT to AI, as viewed from the perspective of AI. We focus on three points:

(i) Human developers use CT to create and develop AI systems (correlating with (b)),
(ii) Understanding how humans learn can enrich AI systems,
(iii) Such enriched systems will be explainable, i.e., they will be able to explain their underlying reasoning (e.g., algorithms) to the end users. This property has often been asked for (e.g., by the EU[1]), but is rarely realized in deep learning approaches.

As regards (i), CT in all its three framings is a *conditio sine qua non* for AI. So far research in AI focuses too much on the end product. It does not adequately regard the process of 'generating' AI methods. It does not yet analyse, how AI methods are developed and to which extent the current AI methods are influenced by our specific ways of CT – the cognitive aspect of CT. Further, as outlined, situativity plays a highly relevant role: we are never developing AI in a sterile, disconnected, independent way, but are instead influenced by

[1] https://ec.europa.eu/futurium/en/system/files/ged/ai-and-interpretability-policy-briefing_creative_commons.pdf

our interactivity and situated learning. Finally, critical assessment of the implications of AI to society, politics, and ethical discussions as an integrative part of developing AI is crucial. Here, dialogue with stakeholders is important, necessitating both a deeper understanding of what AI algorithms can do on the part of stakeholders (without this, their opinions will be superficial at best), and understanding on the part of AI systems developers as regards *how* for instance a political (or ethical or societal) consensus can best be implemented technologically. Ethically, societally and politically appropriate systems development is a back-and-forth movement between stakeholders and AI systems developers.

With regard to (ii), there are several areas where human computational thinking seemingly surpasses AI, and where understanding CT may therefore allow augmenting AI to address known AI problems.

Firstly, there is the *AI frame problem*. The *AI frame problem* concerns the following: if we formalize a state of the world by respective logical formulae and an AI agent performs an action in this world state, the question is, which of the state formulae are changed by the action and which not [see 61]? Despite progress in dealing with the problem from a formal perspective, it still constitutes an important challenge in robotics and in action planning among many other domains. For humans, however, the question is in most cases effortlessly answered – our understanding of the situation means that we know what is changed by an action. Replication and adaption of our world knowledge which frames our understanding of situations has not yet been made available in systems (hence the name: the AI frame problem). Nor has the way we intuitively use this world knowledge in problem solving.

Related to the AI frame problem is the second area, namely *common-sense*. As humans, we very often use common-sense reasoning to help us solve problems, but to date it has not been possible for machines to learn anything like it. Common-sense is therefore considered by Levesque [62] and many others as a core AI problem[2]. For both the AI frame problem and the common-sense problem, coming to understand how we humans develop CT skills across our lifetime may be the solution: It can signpost the way to develop intelligent systems that learn to solve problems like we do. This may be the reason why such different fields as developmental psychology and AI have recently formed an alliance[3].

Third, the literature often conveys the impression that most problem-solving areas can in principle be solved by machine learning. This overlooks the *hierarchy of problem solving* [63] where one must differentiate between at least *permutation problems, insight problems,* and *complex problems*. These differ in the underlying environment (i.e., if it is static or changes), observability (i.e., whether all relevant information is given), and operators (i.e., whether all possible actions are known) among others [64]. While permutation problems, such as playing chess, which has static environments, full observability, and fully known operators, are dealt with fairly well by computers nowadays, this is not the case with insight problems, let alone complex ones. To understand how humans learn to think computationally will provide AI with an approach to higher-level problems that may require more of an understanding than current systems provide. Going back to the three framings above, most AI systems include aspects from the cognitive framing at best, but rarely focus on situativity or critical aspects. In this sense, we can say that AI systems function, but compared to a human that takes other aspects into account they are incomplete.

However, (ii) holds an even greater promise within the field of ultra-strong machine learning [65]. With the rise of connectionist machine learning, it appeared that the capabilities of humans to learn, to abstract, or to develop concepts, need no longer be replicated in AI systems as they have their own way of learning. However, other, symbolically focused, approaches have tried to model the way humans grasp concepts or think generally [66, 67]. If we come to understand *how* humans do CT, we can enrich and augment state of the art AI towards systems that can perform symbolic machine learning – even with the aim of learning the

[2] https://commonsensereasoning.org/problem_page.html

[3] https://www.darpa.mil/program/machine-common-sense

skills humans apply to develop AI systems. This would enable human-centred AI (HCAI), where the control of action is distributed between human and computer with the goal of empowering rather than replacing humans [68]. For instance, if similar reasoning strategies exist between machine and human, a machine learner should be able to teach humans the patterns it identifies and the hypotheses it poses. This would give us (iii): explainable AI, in accordance with Fischer's [69] point that HCAI is an important foundation for explainable AI. A visionary outcome of symbolic AI system development is that systems will be able to conceptualize and themselves develop symbolic AI systems. Such systems would be paradigmatically explainable. Finally, understanding the three framings of human computational thinking allows us to enrich systems with another mode of learning. A learning that takes situativity and the need for critical reflection into account. This may greatly increase the trustworthiness of systems, even as their use as decision-makers for us increases.

Some of the above paragraphs may invoke an *Us versus Them* attitude. This needs to be overcome. It is not CT versus AI. Progress will be made if we can have a true synergy between humans and AI systems, as envisaged by HCAI. But a synergy is only possible if there is a common language for both – and in fact there is one. Humans that learn computational thinking speak the lingua franca of computers and computers already speak that language. The good news is that humans have access to computational concepts, such as loops or case distinction, even if they have not learnt programming constructs yet [70], as also emphasized by the proponents of 'unplugged' CT learning [34].

The chess player Kasparov has envisioned a future where computers and the mind will build a companion system, the human part lends its intuition and the system its brute force capability. In fact, this idea had been proposed much earlier by Michie [71]: "An interesting possibility which arises from the 'brute force' capabilities of contemporary chess programs is the introduction of a new brand of 'consultation chess' where the partnership is between man and machine. The human player would use the program to do extensive and tricky forward analyses of variations selected by his own intuition…" (p. 332). While chess has often been claimed to be the drosophila of AI, there is no reason why this idea should not be applied to problem solving in general – to build AI systems that can connect with humans on a cognitive computational level.

6. Overview of the Special Issue

The preceding paragraphs present a state of the art of CT, with a focus on articulating the relevance of CT for AI. The contributions in the Special Issue move these discussions forward, in terms of offering new theoretical perspectives and novel empirical research on the CT framework and pedagogical approaches presented in Sections 3 and 4, as well as in terms of further explorations of the connections between AI and CT. The Special Issue comprises five articles (technical contributions), one PhD dissertation summary, and two interviews in addition to the present survey. Most of the articles contribute to the field of AI through their investigation of what is involved in learning CT as a prerequisite to learning, developing and critiquing AI (our correlating points (b) and (i) in Section 5). A common denominator of these articles is that they originate in an existing pedagogical approach to learning CT but bring a different theoretical perspective to its study.

The approach of incremental development of problem-solving is the focus of three articles. All three articles challenge the dominant cognitive conceptualization of this approach by applying a situated lens (corresponding more to the situated framing) and investigating problem-solving in its sociocultural and subject-didactical contexts. The first article by is Ane Bjerre Odgaard who investigates how preschool children engage in structured problem-solving activities with tangible digital artifacts (simple age-appropriate robots). Odgaard underscores that from a sociocultural perspective, problem-solving is always a negotiation by participants of how to make concrete sense of the structured tasks in the interaction with things and people in a given context. The second article by Roland Hachmann also investigates the introduction of tangible objects for learning problem-solving techniques, but in contrast to Odgaard's study, the activities are analogue (i.e., take place in the upper left square of Figure 1) and situated in lower secondary school within the new Danish subject Technology Comprehension. Students are given a collaborative escape puzzle task with analogue, tangible objects. Hachmann asks the subject-didactical question how the specific constellation of task, tools and students interact to concretize problem-solving as a Technology Comprehension technique. The third article by Siri Krogh Nordby, Annette Hessen Bjerke, and Louise Mifsud raise a similar subject-didactical question as regards the integration of mathematics and CT in Norwegian primary schools. Through interviews with teachers and classroom observation of both analogue and digital activities (upper left square and lower, middle square in Figure 1), Nordby and colleagues point out that currently CT is viewed by mathematics teachers either as the same as existing mathematics learning practices or, alternatively, as something completely different (an add-on). Full integration, supporting the productive learning of both mathematics and CT – and facilitating students' development of AI literacy – is rare and teachers need help in understanding how this can be done.

The next set of articles examines the connection between AI, CT and pedagogical practice. The fourth article, by Andreas Lindenskov Tamborg, Raimundo Elicer, and Daniel Spikol, investigates the relationship between mathematics and CT, and particularly how the integration of these disciplines can support students in AI learning. This is done with a focus on playful programming and CT (PCT) activities (centering on the lower, middle square in Figure 1). The authors analyze a set of PCT tasks for primary and lower secondary school curriculum in the UK and Denmark, all concerning the building of rock-paper-scissors machines, to highlight how mathematical, PCT competencies and AI learning interrelate. Their emphasis on playfulness shows affinity to the pedagogical take of creative tinkering, but with a narrower disciplinary focus than typical for creative tinkering (represented above in our presentation of makerspaces). The fifth article, by Andrea Valente and Emanuela Marchetti presents a new conceptualization – along with its pedagogical implications – of the approach of iterative problem-solving with IT. As alternative to the cognitive framing usually underlying this approach, the authors present a hermeneutic perspective that combines the hermeneutic spiral with notional machines, supported by a specific library tool, Medialib. From the standpoint that code is text, written in artificial language, the authors conceptualize the use-modify-create progression model as an interpretive process, starting with given code, moving forward to editing and finally creating new code from scratch. They argue that this conceptualization allows simplifying programming for non-technical university students.

The connection between CT, AI and pedagogical practice is highlighted in the articles by Tamborg and colleagues and by Valente and Marchetti in the following ways. Tamborg and colleagues focus on how mathematics can aid students in 'grey-boxing' AI (i.e., in partly opening the black box of AI) through helping them understand fundamental principles and potential implications of AI. Valente and Marchetti call attention to several ways of connecting CT and AI within education, basically arguing that each can support the learning of the other. They point out that teachers of programming typically need support in generating multiple variations of an exercise as well as in validating submitted solutions, and that AI can be developed to help them in both regards. Conversely, they propose that their Medialib tool can be regarded as a high-level recipe for creating domain-specific libraries for beginners and that this can be used in teaching AI to beginners.

The special issue concludes with a PhD dissertation summary and two interviews with leading researchers. The dissertation summary by Anna Keune presents how creative activities with fiber crafts can support students in lower secondary school in their learning of computational concepts. Keune's research illustrates the contextual problem-solving approach and in particular the pedagogical take of creative tinkering. It draws on the situated framing of CT, consistent with the dominant conceptualization of this approach, but adds a further theoretical perspective in investigating the significance of materiality and the forms of learning that various forms of tangibility lead to. The final contributions are two interviews, one with cognitive modeling researcher Sangeet Khemlani, the other with learning scientist Andrea diSessa. In very different ways, they both provide insightful characterizations of the fields of AI, CT and learning – and of their interrelations. Khemlani picks up on the points presented in Section 5 as regards the connection between CT and AI, as viewed from the side of AI, and presents illuminating examples of how CT can augment AI by enabling AI to "play by the same rules that humans play by" (as he puts it). He further points to major challenges in developing cognitive models (and thus of building human-like AI) for learning computational thinking.

DiSessa for his part presents a provocative characterization of “CT as a social movement”, rather than an intellectual enterprise. He points out that as social movement CT is not good or bad per se and suggests we need to consider whether the time spent on learning CT in schools could be better spent on learning something else. He questions the approach of incremental development of problem-solving, its underlying cognitive framing, and in particular the assumption that there exist general thinking skills that are the same across all disciplines. He thus aligns with the situative framing of CT and with the approach of contextual problem-solving.

Declarations

Funding: Research for this article was partly funded by Independent Research Fund Denmark, Grant No. 9130-00006B
Conflicts of interest/Competing interests: None.
Availability of data and material: Not applicable
Code availability: Not applicable

References

[1] A. Newell, J. C. Shaw, and H. A. Simon, "Empirical explorations with the logic theory machine: A case study in heuristics," in *Computers and thought*, E. A. Feigenbaum and J. Feldman Eds. New York: McGraw-Hill, 1963.

[2] K. M. Sayre and F. J. Crosson, *The modeling of mind*. South Bend, Indiana: Notre Dame University Press, 1963.

[3] A. Newell, "Artificial intelligence and the concept of mind," in *Computer models of thought and language*, R. C. Schank and K. M. Colby Eds. San Francisco, California: Freeman, 1973.

[4] S. Russell and P. Norvig, *Artificial intelligence: a modern approach*, 3. ed. (no. Book, Whole). Upper Saddle River, NJ: Prentice Hall (in English), 2010.

[5] S. Papert, *Mindstorms: children, computers, and powerful ideas*. New York: Basic Books, 1980.

[6] I. Goldstein and S. Papert, "Artificial intelligence, language, and the study of knowledge," *Cognitive science,* vol. 1, no. 1, pp. 84-123, 1977.

[7] G. K. W. Wong, X. Ma, P. Dillenbourg, and J. Huan, "Broadening artificial intelligence education in K-12: where to start?," *ACM Inroads,* vol. 11, no. 1, pp. 20–29, 2020, doi: 10.1145/3381884.

[8] J. Wing, "Computational thinking," *Communications of the ACM,* vol. 49, no. 3, pp. 33-35, 2006, doi: 10.1145/1118178.1118215.

[9] J. Wing, "Research Notebook: Computational Thinking - What and Why?," *The Link Magazine,* p. N/A, 2011. [Online]. Available: https://www.cs.cmu.edu/link/research-notebook-computational-thinking-what-and-why.

[10] N. B. Dohn, "Computational thinking - indplacering i et landskab af it-begreber," in *Computational thinking - teoretiske, empiriske og didaktiske perspektiver*, N. B. Dohn, R. Mitchell, and R. Chongtay Eds. Frederiksberg: Samfundslitteratur, 2021, pp. 31-60.

[11] A. van Wynsberghe, "Sustainable AI: AI for sustainability and the sustainability of AI," *AI and Ethics,* 2021/02/26 2021, doi: 10.1007/s43681-021-00043-6.

[12] R. Goebel *et al.*, "Explainable AI: The New 42?," in *Machine Learning and Knowledge Extraction. CD-Make 2018*, A. Holzinger, P. Kieseberg, A. M. Tjoa, and E. Weippl Eds., (Lecture Notes in Computer Science, no. 11015). Cham: Springer International Publishing, 2018, pp. 295-303.

[13] S. Grover and R. Pea, "Computational Thinking in K-12: A review of the state of the field," *Educational Researcher,* vol. 42, no. 1, pp. 38-43, 2013.

[14] A. A. diSessa, *Changing Minds: Computers, Learning, and Literacy*. Cambridge, MA: MIT Press, 2000.

[15] Y. B. Kafai and Q. Burke, *Connected Code: Why Children Need to Learn Programming* (MacArthur Foundation Series on Digital Media and Learning). Cambridge, Mass.: MIT Press, 2014.

[16] J. A. Rode *et al.*, "From computational thinking to computational making," presented at the Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing, Osaka, Japan, 2015. [Online]. Available: https://doi.org/10.1145/2750858.2804261.

[17] M. Tissenbaum, J. Sheldon, and H. Abelson, "From computational thinking to computational action," *Communications of the ACM,* vol. 62, no. 3, pp. 34–36, 2019, doi: 10.1145/3265747.

[18] H. A. Simon, *The sciences of the artificial*, 3 ed. Cambridge, Massachusetts: MIT Press, 1970/1996, pp. xiv, 231 sider.

[19] Y. B. Kafai, "From computational thinking to computational participation in K--12 education," *Communications of the ACM,* vol. 59, no. 8, pp. 26-27, 2016, doi: 10.1145/2955114.

[20] M. Resnick *et al.*, "Scratch: programming for all," *Commun. ACM,* vol. 52, no. 11, pp. 60-67, 2009, doi: 10.1145/1592761.1592779.

[21] E. Hutchins, *Cognition in the Wild* (no. nr. 1995). Cambridge, Massachusetts: MIT Press, 1995.

[22] A. A. diSessa, "Computational Literacy and "The Big Picture" Concerning Computers in Mathematics Education," (in English), *Mathematical thinking and learning,* vol. 20, no. 1, pp. 3-31, 2018, doi: 10.1080/10986065.2018.1403544.

[23] B. Williamson, "Political computational thinking: policy networks, digital governance and 'learning to code'," *Critical Policy Studies,* vol. 10, no. 1, pp. 39-58, 2016, doi: 10.1080/19460171.2015.1052003.

[24] E. Pariser, *The Filter Bubble: What The Internet Is Hiding From You*. London: Penguin Books Limited, 2011.

[25] S. U. Noble, *Algorithms of Oppression: How Search Engines Reinforce Racism* (no. Book, Whole). New York: NYU Press, 2018.

[26] V. Eubanks, *Automating inequality: how high-tech tools profile, police, and punish the poor*, First ed. (no. Book, Whole). New York, NY: St. Martin's Press, 2018.

[27] J. Larson, S. Mattu, L. Kirchner, and J. Angwin, "How we analyzed the COMPAS recidivism algorithm," *ProPublica*. [Online]. Available: https://www.propublica.org/article/how-we-analyzed-the-compas-recidivism-algorithm, accessed July 15, 2021

[28] A. Sfard, "On two metaphors for learning and the dangers of choosing just one," *Educational researcher,* vol. 27, no. 2, pp. 4-13, 1998, doi: 10.2307/1176193.

[29] N. B. Dohn, "Explaining the Significance of Participationist Approaches for Understanding Students' Knowledge Acquisition," *Educational Psychologist,* vol. 51, no. 2, pp. 188-209, 2016. [Online]. Available: http://dx.doi.org/10.1080/00461520.2016.1160321.

[30] J. G. Greeno, "A Situative Perspective on Cognition and Learning in Interaction," in *Theories of Learning and Studies of Instructional Practice*, vol. 1, T. Koschmann Ed., (Explorations in the Learning Sciences, Instructional Systems and Performance Technologies. New York: Springer, 2011, ch. 3, pp. 41-71.

[31] J. G. Greeno and M. S. Gresalfi, "Opportunities to learn in practice and identity," in *Assessment, equity, and opportunity to learn*, P. A. Moss, D. C. Pullin, J. P. Gee, E. H. Haertel, and L. J. Young Eds. New York: Cambridge University Press, 2008, pp. 170-199.

[32] G. Biesta, "Against learning. Reclaiming a language for education in an age of learning," *Nordic Studies in Education,* vol. 25, no. 1, pp. 54-66, 2005.

[33] S. Fox, "An actor-network critique of community in higher education: implications for networked learning," *Studies in Higher Education,* vol. 30, no. 1, pp. 95-110, 2005, doi: 10.1080/0307507052000307821.

[34] E. N. Caeli and A. Yadav, "Unplugged Approaches to Computational Thinking: a Historical Perspective," *TechTrends,* vol. 64, no. 1, pp. 29-36, 2020/01/01 2020, doi: 10.1007/s11528-019-00410-5.

[35] J. J. Lu and G. H. L. Fletcher, "Thinking about computational thinking," *SIGCSE bulletin,* vol. 41, no. 1, pp. 260-264, 2009, doi: 10.1145/1539024.1508959.

[36] Y. Kafai, D. Fields, and K. Searle, "Electronic Textiles as Disruptive Designs: Supporting and Challenging Maker Activities in Schools," *Harvard educational review,* vol. 84, no. 4, pp. 532-556, 2014, doi: 10.17763/haer.84.4.46m7372370214783.

[37] A. Gerdes, "Algoritmisk dannelse," in *Computational thinking - teoretiske, empiriske og didaktiske perspektiver*, N. B. Dohn, R. Mitchell, and R. Chongtay Eds. Frederiksberg: Samfundslitteratur, 2021, in press, ch. 8.

[38] D. Buckingham, "Defining Digital Literacy - What do Young People Need to Know about Digital Media?," in *Digital Literacies: Concepts, Policies and Practices*, C. Lankshear and M. Knobel Eds. New York: Peter Lang, 2008, pp. 73-90.

[39] H. Jenkins, R. Purutshotma, M. Weigel, K. Clinton, and A. J. Robison, *Confronting the challenges of participatory culture: media education for the 21st century* (no. Book, Whole). Cambridge, MA: The MIT Press, 2009.

[40] L. S. Shulman, "Those Who Understand: Knowledge Growth in Teaching," *Educational researcher,* vol. 15, no. 2, pp. 4-14, 1986, doi: 10.3102/0013189X015002004.

[41] R. Chongtay and K. Robering, "Computational Literacy: A Layered Approach for the Humanities," in *9th annual International Conference of Education, Research and Innovation*, Seville, 2016: IATED Academy, pp. 551-561.

[42] I. Lee *et al.*, "Computational thinking for youth in practice," *ACM Inroads*, vol. 2, no. 1*,* pp. 32-37doi: 10.1145/1929887.1929902.

[43] M. Resnick, "Give P's a chance: Projects, peers, passion, play," in *Proceedings of Constructionism and Creativity Conference*, Vienna, Austria, 2014.

[44] M. Eisenberg, "Programmable applications: interpreter meets interface," *SIGCHI Bull.,* vol. 27, no. 2, pp. 68–93, 1995, doi: 10.1145/202511.202528.

[45] M. Resnick, R. Berg, and M. Eisenberg, "Beyond black boxes: Bringing transparency and aesthetics back to scientific investigation," *Journal of the Learning Sciences,* vol. 9, no. 1, pp. 7-30, 2000/01/01 2000.

[46] M. Resnick and E. Rosenbaum, "Designing for tinkerability," in *Design, make, play: Growing the next generation of STEM innovators*, M. Honey and D. E. Kanter Eds. New York: Routledge, 2013, pp. 163-181.

[47] B. Tomlinson, "SUSTAINABLY OURS A call for pro-environmental conspicuous consumption in the online world," *Interactions*, vol. 15, no. 6*,* pp. 42-45doi: 10.1145/1409040.1409051.

[48] K. M. Sheridan, E. R. Halverson, B. K. Litts, L. Brahms, L. Jacobs-Priebe, and T. Owens, "Learning in the making: A comparative case study of three makerspaces," *Harvard educational review,* vol. 84, no. 4, pp. 505-531, 2014, doi: 10.17763/haer.84.4.brr34733723j648u.

[49] G. Skåland, H. C. Arnseth, and P. Pierroux, "Doing Inventing in the Library. Analyzing the Narrative Framing of Making in a Public Library Context," *Education Sciences,* vol. 10, no. 6, p. 158, 2020. [Online]. Available: https://www.mdpi.com/2227-7102/10/6/158.

[50] Y. Baek, E. Min, and S. Yun, "Mining Educational Implications of Minecraft," (in English), *Computers in the schools,* vol. 37, no. 1, pp. 1-16, 2020, doi: 10.1080/07380569.2020.1719802.

[51] R. Sheffield, R. Koul, S. Blackley, and N. Maynard, "Makerspace in STEM for girls: a physical space to develop twenty-first-century skills," *Educational media international,* vol. 54, no. 2, pp. 148-164, 2017, doi: 10.1080/09523987.2017.1362812.

[52] D. Lui, Y. Kafai, B. Litts, J. Walker, and S. Widman, "Pair physical computing: high school students' practices and perceptions of collaborative coding and crafting with electronic textiles," *Computer Science Education,* vol. 30, no. 1, pp. 72-101, 2020/01/02 2020, doi: 10.1080/08993408.2019.1682378.

[53] K. Brennan and M. Resnick, "New frameworks for studying and assessing the development of computational thinking," in *The 2012 annual meeting of the American educational research association*, Vancouver, Canada, 2012, vol. 1, p. 25.

[54] A. R. Basawapatna, A. Repenning, and C. H. Lewis, "The simulation creation toolkit: an initial exploration into making programming accessible while preserving computational thinking," in *The 44th ACM technical symposium on Computer science education*, Denver, Colorado, USA, 2013: Association for Computing Machinery, pp. 501–506, doi: 10.1145/2445196.2445346. [Online]. Available: https://doi.org/10.1145/2445196.2445346

[55] A. I. Mørch, "Two 3D virtual worlds as domain-oriented design environments: closing the educational gap with the action-breakdown-repair model," *The international journal of information and learning technology,* 2020.

[56] A. Basawapatna, K. H. Koh, A. Repenning, D. C. Webb, and K. S. Marshall, "Recognizing computational thinking patterns," in *The 42nd ACM technical symposium on Computer science education*, Dallas, TX, USA, 2011: Association for Computing Machinery, pp. 245–250, doi: 10.1145/1953163.1953241. [Online]. Available: https://doi.org/10.1145/1953163.1953241

[57] G. Fischer, "Domain-oriented design environments," *Automated Software Engineering,* vol. 1, no. 2, pp. 177-203, 1994/06/01 1994, doi: 10.1007/BF00872289.

[58] D. A. Fields, M. Giang, and Y. Kafai, "Programming in the wild: trends in youth computational participation in the online scratch community," in *The 9th Workshop in Primary and Secondary Computing Education (WiPSCE '14)*, New York, NY, 2014: Association for Computing Machinery, pp. 2–11, doi: 10.1145/2670757.2670768. [Online]. Available: https://doi.org/10.1145/2670757.2670768

[59] S. Dasgupta and B. M. Hill, "Scratch community blocks: Supporting children as data scientists," in *Proceedings of the 2017 CHI conference on human factors in computing systems*, 2017, pp. 3620-3631.

[60] S. Hautea, S. Dasgupta, and B. Hill, "Youth Perspectives on Critical Data Literacies," in *CHI '17*, 2017 2017, vol. 2017-, no. Conference Proceedings, New York, NY: ACM, pp. 919-930, doi: 10.1145/3025453.3025823.

[61] J. McCarthy and P. J. Hayes, "Some philosophical problems from the standpoint of artificial intelligence," in *Readings in Artificial Intelligence*, B. L. Webber and N. J. Nilsson Eds. Los Altos, CA: Morgan Kaufmann, 1981, pp. 431-450.

[62] H. J. Levesque, *Common sense, the Turing Test, and the quest for real AI* (no. Book, Whole). Cambridge: MIT Press, 2017.

[63] M. Ragni, "Artificial intelligence and high-level cognition," in *A guided tour of artificial intelligence research*, P. Marquis, O. Papini, and H. Prade Eds. Cham: Springer Nature, 2020, pp. 457-486.

[64] European Schoolnet, "Computing our future - Priorities, school curricula and initiatives across Europe," Brussels, 2014. [Online]. Available: http://www.eun.org/publications/detail?publicationID=481

[65] S. H. Muggleton, U. Schmid, C. Zeller, A. Tamaddoni-Nezhad, and T. Besold, "Ultra-Strong Machine Learning: comprehensibility of programs learned with ILP," *Machine Learning,* vol. 107, no. 7, pp. 1119-1140, 2018/07/01 2018, doi: 10.1007/s10994-018-5707-3.

[66] J. R. Anderson, *How can the human mind occur in the physical universe?* (no. Book, Whole). New York: Oxford University Press, 2007.

[67] P. N. Johnson-Laird, *How we reason* (no. Book, Whole). New York;Oxford;: Oxford University Press, 2006.

[68] B. Shneiderman, "Human-centered artificial intelligence: Reliable, safe & trustworthy," *International Journal of Human–Computer Interaction,* vol. 36, no. 6, pp. 495-504, 2020/04/02 2020, doi: 10.1080/10447318.2020.1741118.

[69] G. Fischer, "End-user development: Empowering stakeholders with artificial intelligence, meta-design, and cultures of participation," in *International Symposium on End-User Development*, D. Fogli, D. Tetteroo, B. R. Barricelli, S. Borsci, P. Markopoulos, and G. A. Papadopoulos, Eds., 2021, Cham: Springer International Publishing, pp. 3-16.

[70] S. S. Khemlani, R. Mackiewicz, M. Bucciarelli, and P. N. Johnson-Laird, "Kinematic mental simulations in abduction and deduction," *Proceedings of the National Academy of Sciences,* vol. 110, no. 42, pp. 16766-16771, 2013, doi: 10.1073/pnas.1316275110.

[71] D. Michie. (1972) Programmer's gambit. *New Scientist*. 329-332.