Chapter 5

Learning Computational Thinking in Phenomenon-Based Co-Creation Projects: Perspectives from Finland

Pasi SILANDER [*]

University of Helsinki, Finland

pasi.silander@hel.fi

Sini RIIKONEN

University of Helsinki, Finland

sini.riikonen@helsinki.fi

Pirita SEITAMAA-HAKKARAINEN

University of Helsinki, Finland

pirita.seitamaa-hakkarainen@helsinki.fi

Kai HAKKARAINEN

University of Helsinki, Finland

kai.hakkarainen@helsinki.fi

Learning Computational Thinking in Phenomenon-Based Co-Creation Projects: Perspectives from Finland

Abstract

The professions and skills needed in modern society are rapidly changing. How will we provide our students with the skills they will need in the future? These *21st-century skills*, such as critical thinking, communication, creativity, and *computational thinking*, cannot be learned through traditional methods. Therefore, there is an urgent need to rethink and redesign education. *Phenomenon-based learning* is one of the most promising new pedagogical approaches and is widely used in schools in Finland. Phenomenon-based learning has been successfully implemented, for instance, in STEAM (science, technology, engineering, arts, math) education and in co-invention projects. In this chapter, we will discuss the relation between phenomenon-based learning, learning computational thinking, and learning *computational creativity skills*. Co-creation and co-innovation will be proposed as metaphors for learning computational thinking and computational creativity skills.

Keywords

Computational thinking, phenomenon-based learning, co-creation projects, 21st century skills, computational creativity skills, STEAM, K–12

Introduction

Societies and industries have changed significantly in recent decades. The emerging *innovation society* has resulted in the technological, sociological, and cognitive development of society. Our professional lives are highly digital, but K–12 education (both teaching and learning) is still taking its first steps in a digital transformation. In order to understand and become an active member of society, students have to learn to understand the technology behind digitalization. Understanding algorithms, such as procedural thinking, reasoning, and decision-making mechanisms, helps students understanding technology and how it works. However, in addition to understanding algorithms and computational thinking, students should be able to utilize them in their personal and collaborative thinking, problem-solving, and creative pursuits.

Modern society relies on advanced technologies, such as artificial intelligence (AI) and data analytics. In order to understand the automatic decision-making of online services and social media, students need computational thinking skills. Moreover, the role of information that is processed and analyzed by AI is increasingly important in our everyday lives. For example, while banking or shopping, a customer's information determines the ads and customized services they see based on automatic decision-making by an AI. When using a search engine or reading a newspaper online, the user is targeted by personalized content and ads based on the motives and content interests of service providers. Learners should be aware that the internet's search engines and social networking tools rate and censor search results and information based on various commercial and political motives.

The major challenge of the K–12 educational system globally is to help students develop critical thinking skills and creative capabilities, especially related to understanding computational processes and mechanisms. In the digital world in which we live, computational thinking skills are a prerequisite for critical thinking. How can we ensure that K–12 educational systems are capable of helping students develop these skills? What methods do we need to use to learn and teach these skills? What wider changes in the organization of teaching and learning in educational institutions are needed?

Computational Thinking as a 21st-Century Skill

Various definitions and frameworks for 21st-century skills (Trilling and Fadel 2009) have been used as a base for K–12 curricula to define transversal competencies and goals for education. Widely used frameworks in K–12 education usually include such competencies as collaboration, communication, citizenship, creativity, critical thinking, and character building. Most 21st-century skills frameworks are focused on so-called soft skills (Bereiter and Scardamalia 2012) and neglect, to a large extent, the importance of logic and mathematical or algorithmic reasoning. Wing (2006) introduced the idea of computational thinking as a fundamental skill for everyone; nevertheless, none of the widely used frameworks have adopted it. Very often, computational thinking is only linked to computer science or STEAM (science, technology, engineering, arts, and mathematics) education and is narrowly understood to only include coding or ready-made mathematical algorithms.'

A common mistake is to talk about coding when we should talk about computational thinking. Coding is often used as a generalized term for programming or, even more often, misused to describe some ill-defined activities with computers. In order to understand how to do programming, it is necessary to comprehend computational thinking and system design. Computational thinking is not a new concept but has been studied and discussed mainly by computer scientists (Wing 2006; Denning 2009; Tedre and Denning 2016). However, it should be more extensively investigated by educational researchers and learning scientists when designing K–12 curricula and educational practices.

The importance of computational thinking was introduced by Wing (2006) and more widely studied by Denning and Tedre (2019). Primitive forms of computational thinking have existed in the form of mathematics and calculation throughout history, even in the time before computers. In modern terms, computational thinking may be defined as cognitive skills and practices for designing computation and computing systems and for explaining and interpreting the world in terms of complex information processes (Denning and Tedre 2019). Wing (2008) has defined computational thinking more compactly as analytical thinking utilizing abstractions, as she defines computing to be the automation of abstractions. However, computational thinking is not only important for computing or for learning programming but it is also a highly generalized cognitive skill needed for critical thinking, media literacy, and knowledge production, as well as for comprehending ethical issues related to data-driven society and various aspects of AI and its ethically sustainable use.

Learning and Teaching Computational Thinking in Modern K–12 Education

The utilization of computational thinking in K–12 education is anchored in our conceptions of emerging digital technology, theories of learning, and technology-mediated practices of learning and teaching. It appears to us that computational thinking requires a new level of epistemic fluency (Markauskaite and Goodyear 2017), interconnecting abstract and real-life phenomena by learners and teachers. When considering pedagogical applications of computational thinking in K–12 education, it is not enough to address mere programming or coding. Programming in K–12 education is sometimes even simplified to routine procedures of giving directions to a computer or to a robot through individual commands. Coding does not equal computational thinking (Wing 2006, 2008) or adequate computing skills; a wider approach than coding is needed for learning and understanding the computational aspects of problem-solving and analyzing, modelling, and automating abstractions (see Figure 5.1). The focus should be on modelling and understanding real-world phenomena by designing, creating, and utilizing abstractions and by creating algorithms and simulations. In addition, the focus of learning should be on systemic thinking, as in system theories or system design.

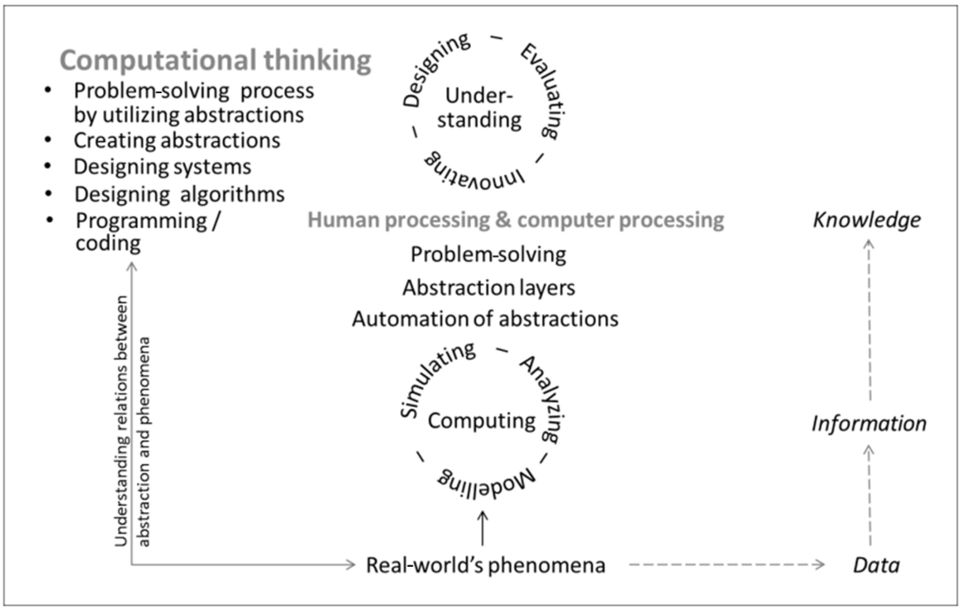

Figure 5.1. Framework for learning computational thinking in K–12 education, consisting of the computational system and human information processing.

Computational thinking skills cannot be adequately learned in a decontextualized setting of programming or designing algorithms without a connection to real-world phenomena and their modelling. We argue that epistemic flexibility is essential to comprehending relations between the real-world phenomena (problems to be solved) and the abstractions (computational models or algorithms) that are used for problem-solving. The goal of learning should be a systemic understanding of the entire computational system, including real-world phenomena, computing, and human information processing.

The use of modern information technology and modern computing are fundamentally culturally mediated cognitive skills. Computational thinking (Wing 2006, 2008) can be associated with metacognitive skills and the sophisticated use of a repertoire of cognitive strategies. Using algorithms as a mental tool augments the power of human cognitive capacity and fosters the development of cognitive strategies. Simultaneously, computing and computers are used as tools for complex physically distributed cognition (Salomon 1993; Pea 1985). Computational power and computers are often used to solve problems that would be difficult or virtually impossible to solve with a human's information-processing capacity. A computational system consists of human cognitive processes, distributed cognition, and information processing on a computer (see e.g., Pea 1985b; Salomon, Perkins, and Globerson 1991), all embedded in the social practices of human communities (see e.g., Ritella and Hakkarainen 2012). Human cognition and computer processing can be seen as intertwined agents of the cognitive system used for complex problem-solving. Moreover, the socially shared cognition mediated by computers boosts these intertwined agents of the cognitive systems that jointly may provide a crucial platform for creating novelty and innovations.

Ideally, when learners are provided with opportunities for cultivating computational thinking skills in K–12 education, they should have generalizable capabilities for organizing, reorganizing, modelling, analyzing, utilizing, and computing information to problem-solve in any subject domain. This raises a pedagogical challenge for K–12 educational systems: how should computational thinking be taught so that students gain adequate skills?

Computational thinking cannot be learned by reading books, by listening to teachers' lectures, or even by coding. Socio-digital processes combined with co-computational thinking are needed. The best way to ensure a holistic understanding of computational thinking (see Figure 5.1) is to connect it to a real-world phenomenon and to pursue complex projects that require the interrelation of concrete experiences with abstractions and associated formal languages. In order to learn novel skills needed for the future, such as computational thinking and creativity skills, new epistemologies (see Table 5.1) and metaphors for learning are needed. Beyond knowledge acquisition, these emerging metaphors of learning highlight the importance of learning through computational participation (Kafai 2016) and collaborative knowledge creation (Paavola and Hakkarainen 2005, 2014). Hence, co-creation and co-innovation are seen as crucial for learning computational thinking and creativity.

Table 5.1. The epistemic approach for learning the traditional and new skills needed in a highly digitalized working life and in modern AI- and data-driven societies.

	Surface learning	**Deep learning**	**Phenomenon-based learning**
Goal	Recalling facts	Understanding	Creating new solutions
Outcome	Capability to apply information only in a narrow context, if at all	Capability to apply knowledge in various situations	Capability to create new solutions for various new situations
Methods	Information acquisition	Collaborative knowledge building	Co-creation and co-innovation
Focus	Facts	Knowledge	Thinking skills and strategies as well as innovation practices

Rather than merely digitalizing traditional acquisition-oriented and teacher-centered instructional practices (surface learning), it is critical to cultivate technology-enhanced practices of learning and instruction that provide opportunities for social participation and collaborative creation of knowledge (Hakkarainen 2009; Paavola and Hakkarainen 2014). In order to appropriate socio-digital instruments as tools of everyday activity, it is necessary to transform everyday practices of learning and instruction as well as change the operational culture of schooling (Ritella and Hakkarainen 2012). Educational transformation is a systemic change and requires strong institutional support to succeed (Fullan 2016; Fullan and Quinn 2015). It is particularly important to develop novel epistemologies of learning and teaching, such as the phenomenon-based approach, in order to integrate the entire community of the school and to promote the pedagogic transformations that the effective learning of computational thinking will call for.

In addition to computational thinking, we propose that computational creativity skills should be a goal of K–12 curricula. We cannot train our children to be merely computer players or even programmers in the future; we will have to train them to become computer composers with real computational creativity skills. To use a musical metaphor, it is not merely about pressing a piano’s keys, but about being able to interpret, compose, and create music. Computational creativity skills are not focused on the automation of existing processes or abstractions of the real world but rather on innovating and creating novel solutions, abstractions, and epistemic artifacts that may not yet exist. *Computational creativity skills* are used to create art and design artifacts, processes, and innovations by using computing, digital fabrication, and shared socio-digital processes.

Phenomenon-Based Learning and Co-Creation Projects as an Approach to Learning Computational Thinking and Computational Creativity Skills Education

Phenomenon-based learning can be described as multidisciplinary inquiry learning where teaching and learning, as well as curriculum, are based on holistic and authentic topics—not on traditional school subjects or decontextualized exercises. The key dimensions of phenomenon-based learning are presented in Table 5.2.

Table 5.2. Key dimensions of phenomenon-based learning.

Holism	The topics and concepts to be learned are chosen for their relevance in the real world, and a 360° perspective is offered through the integration of traditional school subjects.
Authenticity	The methods, tools, materials, and cognitive practices used in learning situations should correspond to ones in the real world: for example, in professional life.
Contextuality	Learners learn new things in their natural context and learn to move fluidly between contextualization and abstraction.
Problem-based inquiry learning	Learning and collaborative knowledge building are based on the questions and problems posed by learners, and solutions are created by them as well, allowing them to take an active role in designing the curriculum.
Learning as a nonlinear process	Learning is seen as a nonlinear process, which is activated, guided, and facilitated by open learning challenges and supporting structures.

The basis of phenomenon-based teaching and learning can be found in constructionism, which sees learners as active builders and creators of knowledge and artifacts. Knowledge is constructed as a result of problem-solving and creative production through the integration of little pieces into a comprehensive whole according to the situational needs and the information available at the time. When phenomenon-based learning occurs in a collaborative setting (when the learners work in teams, for example), it supports the socio-constructivist and socio-cultural learning theories, in which knowledge is not merely an internal element of an individual. Instead, knowledge is formed in a social context. Socio-cultural learning theories focus on cultural artifacts (e.g., systems of symbols, such as language, mathematical calculation rules, and different kinds of thinking tools). Learning relies on the knowledge and tools, which are transmitted by cultures, that are used generatively in novel contexts and for novel purposes.

Phenomenon-based learning begins with the shared observation of holistic, genuine real-world phenomena in the learning community. The phenomena are studied as complete entities in their real context, and the knowledge and skills related to them are studied by crossing the boundaries between school subjects. Phenomenon-based integrative study units frequently represent such holistic topics as climate change, the water cycle, and health and nutrition. This differs from traditional school culture, which is divided into subjects, where the things studied are often split into relatively small, separate, and decontextualized parts.

In phenomenon-based teaching, understanding and studying the phenomenon starts by asking a question or posing a problem (e.g., why does an airplane fly and stay up in the air?). At its best, phenomenon-based learning is cyclic inquiry learning, where the learners ask questions or pose problems about a phenomenon that interests them and then discover answers and find solutions together. The problems and questions are posed by the learners together—they are things the learners are genuinely interested in. Learners play a central role in creating and solving the learning challenges being pursued.

The observation is not limited to a single point of view; instead, the phenomena are studied from various points of view, crossing the boundaries between school subjects naturally and integrating subjects like mathematics, history, foreign languages, and psychology with a variety of themes. Phenomenon-based structure in a curriculum actively creates better opportunities for integrating computational thinking in various subjects and themes and for the systematic use of pedagogically meaningful methods, such as collaborative knowledge building (Scardamalia and Bereiter 2006), flipped classrooms (see e.g., Bergmann and Sams 2012), and computational participation (Kafai 2016). The phenomenon-based approach is also key to the versatile utilization of various digital learning environments (e.g., diversifying and enriching learning while using online learning environments).

In the learning process, new knowledge and skills are always applied to the phenomenon or the problem at hand, which means that the concepts, knowledge, and skills have immediate utility value that is evident in the learning situation. In order to absorb new knowledge and skills, it is very important that learners apply and use the knowledge and skills, such as computational thinking, during the learning situation. Information learned only at the level of reading or theory (such as memorized physics formulas and calculation rules without real context or related problems) often remain superficial and separate details for the learners. They are unable to gain a comprehensive understanding and deeper knowledge of the real-world phenomenon and unable to internalize its meaning. Often, it has been said that "you cannot learn to drive a car by using pen and paper" or that "cloze tests only teach how to answer cloze tests—there are no cloze tests in real life or professional life." Beyond encapsulated schoolwork, there are real communication situations where knowledge must be applied and messages must be transmitted clearly and comprehensively to another person.

The phenomenon-based approach can significantly increase the authenticity of learning. This authenticity culminates in making the learner's cognitive processes and practices authentic. In a learning situation, the learner's cognitive processes, therefore, correspond to the cognitive practices required in the actual situation in which the knowledge and skills would be used. Toward that end, it is important to engage learners in creative activities that guide them to adopt the practices and epistemic games (Shaffer and Gee 2007) of computer scientists, designers, engineers, and scientists. In this authentic learning, the aim is to bring genuine practices and processes into learning situations in a pedagogically structured way when applicable, which allows the learner to participate in the expert culture of the field. Authenticity is a key requirement for the transfer and practical application of knowledge.

The new phenomenon-based approaches for teaching and learning computational creativity skills are fostered by the novel affordances of socio-digital technologies that provide sophisticated professional-level tools for creative production. Associated practices involve, for instance, students learning by designing and building robots or utilizing 3D HoloLens, 3D printers, and sensors in their creative projects. The phenomenon-based projects emphasize a way of thinking in which students solve authentic design challenges thorough various collaborative design activities, apply computational thinking, and do actual coding, depending on the nature of the project.

Many Finnish schools are building educational makerspaces (see e.g., Peppler, Halverson, and Kafai 2016) by integrating arts and crafts, technology education, and science laboratories into other school subjects. Schools in Helsinki have organized codesign and co-invention projects that engage learners in designing complex artifacts that spark intellectual, engineering, and aesthetic challenges at lower and upper primary schools (Seitamaa-Hakkarainen and Hakkarainen 2017). Students work in small teams to solve an open-ended invention challenge using traditional craft and digital fabrication technologies. Their projects, in which they create various prototypes and products that assist in modelling the phenomenon, test and develop the learners' hypotheses and working theories. The challenge, which is co-configured with learners, might be, for example, to "design an intellectually challenging, aesthetically appealing, and personally meaningful complex artifact that makes daily tasks easier." It could be a new or an improved invention, and it should integrate both physical and digital (e.g., circuits or robotic) elements.

The role of teachers is not merely to facilitate learning but also to activate students' computational thinking and learning processes. Toward that end, the learning-by-making activities are structured according to several stages, including skill building (e.g., working with microcontroller or other circuit boards), orientation (guided analysis of existing artifacts), and brainstorming with design challenges (in the classroom and at home with parents). They analyze design constraints (task requirements and resources), cluster design ideas, identify promising ones, and decide on their teams' design project. They share design ideas in the classroom, get feedback, seek knowledge (e.g., by visiting technical or design museums), experiment with design solutions, and construct prototypes of the design to arrive at their final solutions. It is also very important to organize exhibitions where teams can present their co-inventions to other students and parents. The analysis of Sinervo et al. (2020) of the designs of thirteen fifth-grade students (aged eleven to twelve years old) revealed that the details of their innovations varied considerably. We categorized the teams' co-inventions according to their main function, such as improving cleanliness, providing reminders, or addressing hygiene, health, and nutrition issues. The inventions also reflected issues related to user values (health-related inequality, inclusion, or personalization), use values (helping to resolve problematic situations), and environmental values (Sinervo et al. 2020).

Most of the teams' co-inventions were considered appropriate and promising, and only two co-inventions were not explicated clearly enough and could be considered quasi-creative and infeasible. Some very original ideas for known problems were found—for example, how to vacuum a carpet and the creation of a new gel comb for styling hair, even though these teams were not able to construct fully functional solutions. The gel comb team had a hard time figuring out how to get the gel out of the container. Some of the co-inventions were based on an already existing idea or product that was used in another context—for example, a pump bottle that was extended to help brush teeth with toothpaste more easily. In some cases, the co-invention was based on the adaptation of existing artifact designs by slightly modifying an existing product—for example, an automatic garbage container with an alarm that sounds when it is almost full. This long-term, open-ended invention project provided valuable learning opportunities for iterative problem-solving, shared meaning making, and collaboration that required a division of labor, organization, and personal responsibility. Phenomenon-based learning empowers students to participate in the co-creation and co-innovation processes that are needed to learn computational thinking skills and computational creativity skills. By using co-creation and co-innovation as learners' activities, the learning process is more insightful and inspiring. The role of the learner is not that of an object but that of an active subject of learning.

A more demanding example of a phenomena-based co-invention project was conducted with one class of seventh-grade students (aged thirteen to fourteen). The project was initiated by the craft and visual arts teachers and involved the participation of mathematics, physics, chemistry, and information technology (IT) teachers, who provided their expertise to the inventors when needed. Eighth-grade digital technology students who had done a similar project the year before also helped the inventors during the project. The project started with two warm-up sessions for skill building. In the first session, the students built electric circuits using cards with copper tape, simple LEDs, and a coin cell battery. The aim of this warm-up session was to familiarize the students with basic electric circuits, so they would be able to use them in their inventions. The second warm-up session was organized by the eighth-grade students; they planned and held a workshop for the seventh graders about microcontrollers, basic programming with block-based coding, sensors, and DC motors. Many of the students had only done very simple Scratch programming tasks before this. After that, the actual collaborative invention project was initiated, and it ran for eight to ten weekly two-hour sessions. Also, in this project, the collaborative invention challenge was open-ended: “Invent a smart product or a smart garment by relying on traditional and digital fabrication technologies, such as microcontrollers and 3D CAD.” At the end of the project in May, the teams presented their inventions in an open invention exhibition held at the University of Helsinki.

This project proceeded much as the previous example had; it was initiated and led by the student teams. The teachers and tutors provided help when needed, but the project teams took most of the responsibility for the design and the construction. As the challenge required, student teams needed to use various digital technologies. It was also typical of the teams’ processes that while ideating and experimenting, they confronted many phenomena related to physics, such as mechanics, electronics, and light and optics. Thus, they were exposed to numerous physics principles without being necessarily conscious of it. For example, one team (the banana light team) invented a banana-shaped LED light that attaches to a laptop lid and lights up the keyboard area. The features of their lamp included an RGB LED controlled by a microcontroller and a bendable structure that allowed the light to be directed to the keyboard.

During their design process, the team produced sixty-three design ideas in total, which can be divided into seven themes: (1) aesthetic features and name of the project; (2) materials; (3) light controls; (4) mounting to the laptop; (5) electrical connections; (6) directing light; and (7) other functions. The banana light team's invention process had many science-intensive steps. For example, when the team designed the structure of the lamp, some concepts of mechanics became relevant. With the joints, they experimented intensively with 3D models and concrete prototypes. While searching for ways to attach the lamp to the lid of the laptop, the concept of friction came up. Furthermore, as the light was the main functionality of their invention, they spent a lot of time designing it and, thus, light and optics concepts were studied many times during the team's work. The microcontroller was used to operate the LED lights of the invention, and they tested several different options for controlling them, especially for turning them on and off. Understanding classical IF logic was particularly significant in these experiments in terms of learning programming and basic computational thinking. Figure 5.2 shows a sketch and prototype of the banana light.

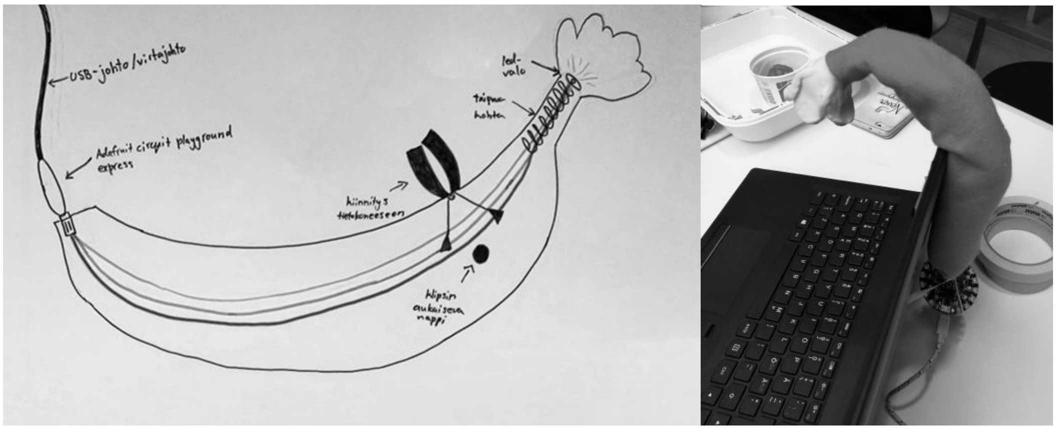

Figure 5.2. Illustrating, designing, and making a prototype of the banana light invention.

Furthermore, the team continued by testing different methods of turning the light on and off with predetermined event functions of the microcontroller, such as tapping the microcontroller twice or clapping their hands to create a loud sound. In the second prototype, they ended up using a simple button that they determined would be the most reliable when presenting the lamp to an audience in a noisy environment. Later, they decided to take their programming a bit further and added a functionality to control the brightness of the LED with the board's second button. The team was able to design a fully functional prototype meeting their specifications. These and other extensive maker-centered learning projects allow students to build epistemic flexibility in terms of interrelating concrete and abstract phenomena and, thereby, provide ample opportunities for system design and learning computational creativity and computational thinking skills.

Discussion

The activities of computational thinking and programming are not equivalent to a human giving commands to a computer or a robot. Instead, they involve problem-solving and creativity, enhanced with computational tools and languages. It is not a matter of mastering certain commands or coding procedures but of engaging a designing system and creating digitally enhanced artifacts. How can we transform the educational system to help transform children from computer players to digital makers with real computational creativity skills?

In order to succeed in modern society, students should have advanced socio-digital and computational thinking skills when they complete their K–12 education. These essential skills are needed across all fields of study, from the humanities to the sciences, including productive participation in knowledge-intensive work, and for becoming an active citizen in data- and AI-driven digital societies. Computational thinking cannot, however, be learned incidentally, for example, by playing computer games or by coding at home. Although informal interest-driven and creative participation is important for overcoming digital divides, formal education that deliberately cultivates innovative pedagogy and the associated teachers' expertise and guidance are urgently needed as well. The best way to provide computational thinking skills and computational creativity skills for all students is to integrate them into K–12 education in curricula and in everyday teaching and learning practices in the form of phenomenon-based co-creation projects. As we live in highly digital societies, we should also start discussing 22nd-century skills, which will be focused on the innovation skills needed in an emerging innovation-driven society that is thoroughly based in AI and the smart use of big data. Computational thinking and computational creativity skills are the key competencies of such a society's citizens.

Practical Implications for Curriculum Design and for Educational

Institutions

Learning computational thinking should begin from early childhood (for example, in the form of cognitive games, songs, and plays) and continue across the whole span of education. Digital technologies develop expansively and continuously, so the process of learning computational thinking and computational creativity skills should also be a sustaining, lifelong learning process. A significant challenge of teacher education is to help teachers develop digital and computational thinking skills that they did not have the opportunity to learn during their own childhood education. Only by acquiring computational skills and practices can teachers work as builders of children's futures. In order to teach computational thinking and creativity skills in K–12 education, both competent and educated teachers and the context and time for cultivating such competencies in teaching and learning are urgently needed. This creates a challenge for teachers' in-service training. How can teachers be trained in pedagogical skills and methods that will scaffold students' computational thinking and computational creativity skills? Our experiences indicate that novel professional competencies become accessible when teachers are encouraged to collaborate with their colleagues and negotiate challenges through co-teaching. Teacher training should be thoroughly participatory and should engage teachers in co-creation and co-invention projects similar to those of young learners.

Traditional computer science and programming education do not offer ready-made solutions for learning computational thinking or computational creativity skills in K–12 education. Instead, new practices and innovations require new pedagogical considerations in educational institutions on the level of the curriculum. An optimal impact on computational thinking with phenomenon-based learning and co-creation projects can be achieved by implementing the change comprehensively throughout the school's operating culture and by ensuring that computational thinking and phenomenon-based learning are integrated into the holistic reform of teaching and learning. The challenge is to implement the pedagogical change coherently and simultaneously at all levels (teaching, leadership, learning, technology, and curriculum). According to Fullan (2016), system improvement will result from a deep change in the culture of learning, local ownership of the learning agenda, and a system of continuous improvement and innovation that is simultaneously bottom-up, top-down, and sideways. Through systemic developmental efforts that integrate all levels, a permanent change in the operating culture can be achieved.

Acknowledgments

This research was supported by the Academy of Finland (Grant 286837) and Strategic Research Grant 312527 of the Academy of Finland (Growing Mind Research Project).

References

Bereiter, Carl, and Marlene Scardamalia. 2012. "What Will It Mean to Be an Educated Person in Mid-21st Century?" Unpublished manuscript. https://www.ets.org/Media/Research/pdf/bereiter_scardamalia_what_will_mean_educated_person_century.pdf

Bergmann, Jonathan, and Aaron Sams. 2012. *Flip Your Classroom: Reach Every Student in Every Class Every Day*. Washington, DC: International Society for Technology in Education.

Bryk, Anthony S., Louis M. Gomez, and Alicia Grunow. 2010. "Getting Ideas into Action: Building Networked Improvement Communities in Education." Carnegie Perspectives. Carnegie Foundation for the

Advancement of Teaching. http://www.carnegiefoundation.org/spotlight/webinar-bryk-gomez-building-networked-improvement-communities-in-education.

Denning, Peter J. 2009. "The Profession of IT Beyond Computational Thinking." *Communications of the ACM*, *52* (6): 28–30.

Denning, Peter J., and Matti Tedre. 2019. *Computational Thinking*. Cambridge, MA: MIT Press.

Fishman, Barry J., William R. Penuel, Anna-Ruth Allen, Britte Haugan Cheng, and Nora Sabelli. 2013. "Design-Based Implementation Research: An Emerging Model for Transforming the Relationship of Research and Practice." *National Society for the Study of Education*, 112, 136–56.

Fullan, Michael. 2016. "The Elusive Nature of Whole System Improvement in Education." *Journal of Educational Change*, 17 (4): 539–44.

Fullan, Michael, and Joanne Quinn. 2015. *Coherence: The Right Drivers in Action for Schools, Districts, and Systems*. Thousand. Oaks, CA: Corwin Press.

Hakkarainen, Kai. 2009. "Three Generations of Technology-Enhanced Learning." *British Journal of Educational Technology*, 40 (5): 879–88.

Kafai, Yasmin B. 2016. "From Computational Thinking to Computational Participation in K-12 Education." *Communications of the ACM*, 59 (8): 26–7.

Markauskaite, Lina, and Peter Goodyear. 2017. *Epistemic Fluency and Professional Education: Innovation, Knowledgeable Action and Actionable Knowledge*. London: Springer.

Paavola, Sami, and K. Hakkarainen. 2005. "The Knowledge Creation

Metaphor: An Emergent Epistemological Approach to Learning." *Science and Education*, *14*, 535–57.

Paavola, Sami, and Kai Hakkarainen. 2014. "Trialogical Approach for Knowledge Creation." In *Knowledge Creation in Education*, edited by S. C. Tan, H. J. Jo, and J. Yoe, 53–73. Singapore: Springer.

Pea, Roy D. 1985. "Integrating Human and Computer Intelligence." In *New Directions for Child Development, No. 8, Children and Computers*, edited by Elisa L. Klein, 75–96. San Francisco: Jossey-Bass.

Pea, Roy D., D. Midian Kurland, and Jan Hawkins. 1985b. "Logo and the Development of Thinking Skills." In *Children and Microcomputers: Formative Studies*, edited by Milton Chen and William Paisley, 193–212. Beverly Hills, CA: Sage.

Peppler, Kylie, Erica Halverson, and Yasmin B. Kafai. 2016. *Makeology: Makerspaces as Learning Environments*, vol. 1. London: Routledge.

Ritella, Giuseppe, and Kai Hakkarainen. 2012. "Instrument Genesis in Technology Mediated Learning: From Double Stimulation to Expansive Knowledge Practices." *International Journal of Computer-Supported Collaborative Learning*, 7, 239–58.

Salomon, Gavriel. 1993. *Distributed Cognitions: Psychological and Educational Considerations*. Cambridge, MA: Cambridge University Press.

Salomon, Gavriel, David N. Perkins, and Tamar Globerson. 1991. "Partners in Cognition: Extending Human Intelligence with Intelligent Technologies." *Educational Researcher*, 20 (3): 2–9.

Scardamalia, Marlene, and Carl Bereiter. 2006. "Knowledge Building: Theory, Pedagogy, and Technology." In *Cambridge Handbook of the Learning*

Sciences, edited by Keith Sawyer, 97–118. New York: Cambridge University Press.

Seitamaa-Hakkarainen, Pirita, and Kai Hakkarainen. 2017. "Learning by Making." In *The SAGE Encyclopedia of Out-of-School Learning*, edited by Kylie Peppler. Thousand Oaks, CA: Sage.

Shaffer, David Williamson, and James Paul Gee. 2007. "Epistemic Games as Education for Innovation: Learning through Digital Technologies." *BJEP*, Monograph Series II, 5, 71–82.

Sinervo, Stiina, Kati Sormunen, Kaiju Kangas, Kai Hakkarainen, Jari Lavonen, Kalle Juuti, Tiina Korhonen, and Pirita Seitamaa-Hakkarainen. 2020. "Elementary School Pupils' Co-Inventions: Products and Pupils' Reflections on Processes." *International Journal of Technology Design and Education*, https://doi.org/10.1007/s10798-020-09577-y.

Tedre, Matti, and Peter J. Denning. 2016. "The Long Quest for Computational Thinking." In *Proceedings of the 16th Koli Calling International Conference on Computing Education Research*, 120–9. New York, NY: ACM.

Trilling, Bernie, and Charles Fadel. 2009. *21st Century Skills: Learning for Life in Our Times.* New York: John Wiley.

Wing, Jeannette M. 2006. "Computational Thinking." *Communication of the ACM*, *49* (3): 33–5.

Wing, Jeannette M. 2008. "Computational Thinking and Thinking about Computing." *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, no. 1881: 3717–25.