Computers and Education Open 4 (2023) 100132

Contents lists available at ScienceDirect

Computers and Education Open

journal homepage: www.sciencedirect.com/journal/computers-and-education-open

Student and teacher co-agency when combining CT with arts and design in a cross-curricular project

Linda Mannila [a,*], Teemu Leinonen [b], Merja Bauters [c], Marjaana Veermans [d]

[a] Dept. of Computer and Information science, Linköping University, SE-581 83 Linköping, Sweden
[b] Media Lab, Department of Media, Aalto University, P.O. Box 11000, FI-00076 AALTO, Finland
[c] School of Digital Technologies, Tallinn University, Narva rd 25, 10120 Tallinn, Estonia
[d] Department of Teacher Education, University of Turku, FI - 20014 Turun Yliopisto, Finland

ARTICLE INFO

Keywords:
K-12 education
Computational thinking
Cross-curricular projects
Arts and design

ABSTRACT

The technological development has raised awareness for the importance of digital competence and computational thinking (CT) to understand the digital world and has resulted in revised curricula in many countries. In Finland, a new curriculum for grades 1–9 came into force in 2016 introducing digital competence (including programming) to be integrated in other subjects. Most teachers lack prior experience in programming and there is a need for suitable instructional models. This article presents a cross-curricular teaching sequence and the results from a case study conducted in four Finnish schools. Students in grades 4–6 collaboratively worked on a project combining arts, design and CT with other subjects. The results show that students demonstrated several CT abilities while working on their projects, in particular creativity, tinkering and debugging. The findings also indicate that teachers and students learned together (co-agency) and suggest that models like the teaching sequence can help and encourage teachers to integrate programming and CT in a cross-curricular manner. Still, the teachers' knowledge, ambition level and understanding of the task at hand, as well as the organizational support appear to play a notable role when planning and carrying out projects of this kind. While CT is commonly seen as developed through programming, the teaching sequence seems to have fostered CT abilities through the project as a whole, with programming playing the role of a tool or a glue depending on the time available, and the students' skill and ambition level.

1. Introduction

The rapid technological development affects all societal areas and our everyday lives. As a result, voices have been raised globally for the importance of computer science and programming [30] and digital competence [93] for all. The rationale is that – in a digital world – everyone needs to understand the basics of the underlying technology as well as the opportunities and potential challenges it brings.

Instead of mere programming, there has been an increased discussion about computational thinking (CT) as a fundamental set of concepts, approaches and attitudes for solving problems with the help of computers [96,102]. CT should hence not merely be considered a new body of knowledge but also seen as essential practices needed in a complex and increasingly digital world. As Schleicher [82] points out, success in education is no longer only about content knowledge but rather epistemic knowledge, as "*thinking like a scientist, philosopher or mathematician – is taking precedence over knowing specific formulae, names or places*" (p. 31). This is also reflected in a joint report from ACM Europe and Informatics Europe [90], highlighting that the *"foundational principles of [Computer Science] and its characteristic ways of shaping thinking, expression, and work are more important for education than its specific technologies"* (p. 6).

Similar ideas have been put forward in other contexts, and as a result, many countries have renewed their curricula for K-12 education, introducing CT [20,21,28,104]. The new content has been introduced in different ways, not necessarily explicitly as CT, but rather as digital competence, computing or programming. While some countries have introduced a completely new subject, others, such as Finland, have integrated the new content into existing subjects. The integration has been accomplished by, for example, introducing the new content as an interdisciplinary element throughout the curriculum or as part of a few subjects [7].

* Corresponding author.
E-mail address: linda.mannila@liu.se (L. Mannila).

https://doi.org/10.1016/j.caeo.2023.100132
Received 29 June 2021; Received in revised form 3 February 2023; Accepted 3 March 2023
Available online 21 March 2023

Even though CT is a common topic in K-12 education, there are no accepted standards for how to teach it. Programming is, however, commonly recognized as a key element for improving students' CT skills [32,57]. Consequently, introducing CT through programming seems to be the prominent approach [7,105].

In Finland, a new national curriculum for grades 1–9 (approximately ages 7–15) was accepted in 2014, coming into force gradually starting in 2016. The curriculum does not explicitly mention CT but introduces programming as part of the cross-curricular theme digital competence. In addition, programming is explicitly mentioned in two subjects: mathematics and sloyd.[1]

The introduction of CT in national K-12 curricula has placed teachers worldwide in a challenging position, as they are expected to teach a topic that is completely new to most of them. As a result, both international (e.g., [79,92,106]) and Finnish (e.g., [86,107]) studies have shown that teachers struggle with integrating programming in their classrooms due to, for instance, lack of skills, time and professional development.

In this paper, we present a model used to introduce CT in a cross-curricular manner at middle school level in Finland. The model builds on an art and design approach and has been empirically evaluated in a case study involving teachers and students at four Finnish schools. By analysing student projects as well as teacher and student learning diaries, we bring light on both teachers' and students' experiences as well as the CT abilities students demonstrated. While there is a growing amount of research on CT in K-12 education, as discussed below, most empirical studies focus on, for instance, tools, curriculum or assessment. Our goal was rather to get insight into what aspects of CT arise from the students' projects and their work descriptions. In addition, we wanted to investigate how the student results related to the teachers' experiences to bring light on how teachers transform their instruction based on the curriculum and thereby build agency in teaching CT. The main research questions addressed in the study were hence the following:

- RQ1: How can a cross-curricular teaching sequence covering CT be designed for middle school level?
- RQ2: How do teachers approach such a teaching sequence and build agency to implement it in the classroom?
- RQ3: What aspects of CT do students engage in or experience when participating in such a project?

The rest of the article is organized as follows. To set the stage for the study, we first present the background and some key concepts. Next, we present the study settings and method, including the cross-curricular teaching sequence, data collection, analysis frameworks and ethical considerations. Next, the findings are presented and discussed, after which the article is concluded with some words on limitations and future ideas.

2. Background

2.1. Defining and operationalizing CT

CT was coined already in 1980 by Papert [68], who described it primarily as the relationship between programming and thinking skills. He believed that students' constructions when programming in LOGO could facilitate their procedural thinking across multiple disciplines. CT gained renewed interest through a seminal article by Wing [96], suggesting that CT is *"a universally applicable attitude and skill set everyone, not just computer scientists, would be eager to learn and use"* (p. 33). Wing defined CT as *"solving problems, designing systems, and understanding human behavior, by drawing on the concepts fundamental to computer science."* (2006, p. 33) and argued that CT, *"just like reading, writing, and arithmetic, should be added to every child's analytical ability"* (p. 33). Wing later updated the definition to view CT as "*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*" [97].

In addition to Wing's theoretical definitions above, other organizations have tried to operationalize CT. For instance, the International Society for Technology in Education (ISTE) included CT as one of the seven main standards describing skills and qualities needed to enable students to *"thrive in a constantly evolving technological landscape"* already in 2016 [43]. According to the standards put forward for CT, the goal is for students to learn how to *"develop and employ strategies for understanding and solving problems in ways that leverage the power of technological methods to develop and test solutions."*

Another way of operationalizing CT is through models of CT concepts and practices. Already in 2011, the American Computer Science Teachers Association (CSTA) and the International Society for Technology in Education (ISTE) proposed a definition identifying nine concepts: data collection, data analysis, data representation, problem decomposition, abstraction, algorithms, automation, parallelization, and simulation [16]. Also in 2011, Barr and Stephenson [4] described how these nine core computational thinking concepts could be integrated into different subject areas. A year later, Brennan and Resnick [9] proposed a model describing CT in terms of three dimensions: concepts (sequences, loops, events, parallelism, conditionals, operators and data), practices (experimenting and iterating: testing and debugging; reusing and remixing, and abstracting and modularizing) and perspectives (expressing, connecting, and questioning). This framework has later been extended by Zhang and Nouri [103] to include skills as a fourth dimension.

Yet another example of this broader way of operationalizing CT is the definition used by Barefoot Computing [3] in England, which builds on seven concepts (logical thinking, algorithms, decomposition, pattern recognition, abstraction and evaluation) and five approaches (tinkering, creating, debugging, persevering and collaborating). The above concepts and skills were synthesized by Fagerlund et al. [25] into a list of 14 core educational principles describing the fundamental skills and areas of understanding involved in CT. Tikva and Tambouris [88] presented a conceptual model of CT in K-12 education. Through an analysis of 101 relevant studies, they found six CT areas: knowledge base, assessment, learning, factors, tools, and capacity building. The first one, knowledge base, was identified as the core; over half of the studies represented this CT area. This area could be further divided into five sub-areas: concepts, skills, practices, perspectives, and skills.

2.2. Introducing CT in the K-12 classroom

In a review of studies on CT in K-12 education, Shute et al. [83] found no consensus on how to best teach CT, but rather a variety of definitions, assessment methods, interventions and models. Nevertheless, as mentioned in the introduction, CT is commonly introduced using programming [7,57]. For instance, Hsu et al. [40] found that visual programming languages were the most common tool for teaching CT, while Tang et al. [85] observed that most research on CT assessment focused on students' programming skills. They also noted that many researchers define CT based on programming and computing concepts.

The relationship between programming and CT has been explained in different ways. For instance, Grover and Pea [32] noted that programming can both support the cognitive tasks involved in CT and be used to demonstrate computational competencies. Similarly, Tikva and Tambouris [88] point out that there is a dual association between the two: programming offers the mechanisms needed for implementing CT, while CT gives programming a new role as a way to explain the world. In their review from 2021, Sun et al. [84] concluded that programming education can cultivate K-12 students' CT skills.

[1] Sloyd is a subject taught in the Nordic countries and can be described as crafts or handicraft using both soft and hard materials.

Studies on CT in K-12 education focus on a range of aspects. Curriculum design (e.g., [35,94]) and teaching approaches (e.g., [20,108]) are naturally central areas when introducing new content in education. Another common research topic focuses on the tools used, such as unplugged programming [5,10,17]), the programming environment Scratch (e.g., [25,64,103]), robotics (e.g., [2,13]) or microcontrollers (e.g., [27]).

As CT is a multidimensional concept, researchers have also studied how programming activities foster CT among students. Fagerlund et al. [25] found that some core educational principles (efficiency, data, abstraction and automation) are not as contextualized in students' Scratch programs as others. Arfé and others [1] concluded that programming practice can improve problem-solving and planning skills among children as young as six years old. Israel-Fishelson and colleagues [42] discovered associations between CT and two aspects of creativity: creative thinking and computational creativity.

Another popular research topic is CT assessment. Many efforts are based on a given context (e.g., [24]), while others aim at developing more general assessment methods, for instance, in the form of tests [12,78], rubrics [62] and instruments for getting insight into, e.g., students' thought processes [89], attitudes [51] and experienced self-efficacy [53]. Grover [31] argued that CT cannot be assessed based on one single measure, but rather calls for "multiple measures or 'systems of assessments' that are complementary, attend to cognitive and noncognitive aspects of learning CT, and contribute to a comprehensive picture of student learning" (p. 269). One attempt at such a framework is the *PESS* instrument building the assessment on previous experience, self-efficacy and skills [109].

When CT is introduced in education, it is not sufficient to only focus on the students, but the teacher's perspective must also be considered. This is reflected in previous research, as many studies cover the teachers' point of view. For instance, already in 2014, an international working group reviewed the current state of CT in K-9 education, suggesting that some teachers were already engaged in CT-related teaching activities [110]. Teacher perceptions of and attitudes towards CT have also been studied (e.g., [81]), while other researchers have focused on professional development for in-service teachers (e.g., [51]), pre-service teachers (e.g., [29,54]) and the evaluation of such training initiatives (e.g., [76]).

2.3. STEAM and maker culture

CT is not an isolated skill set but includes skills that can be used to solve problems in a particular context or knowledge area. Consequently, researchers have explored the transversal potential of combining CT with other subject areas, such as science and mathematics (e.g., [58,95]), languages (e.g., [44,69]), geography (e.g., [33]), music (e.g., [74,80]) as well as arts and literature (e.g., [19]).

One area in which CT is commonly seen as a natural component is STEM (science, technology, engineering and mathematics), as CT is considered the "connecting tissue" between computing and disciplinary knowledge [60]. For instance, CT is believed to have the potential to deepen students' STEM learning by letting them engage in authentic practices [55].

STEM has traditionally been considered an essential part of education in many countries as the subjects are commonly seen as the source of innovation and ground-breaking research. During the last decade, however, we have seen an increasing shift in the discussion from STEM to STEAM, that is, STEM complemented with arts. Already in 2010, Piro argued that "if creativity, collaboration, communication, and critical thinking – all touted as hallmark skills for 21st century success – are to be cultivated, we need to ensure that STEM subjects are drawn closer to the arts" [75], p. 29. Both STEAM and arts and design projects are multidisciplinary in their nature and promote learning through and with other disciplines [72]. Integrating arts with STEM subjects can also encourage creativity and more expansive domain learning [34]. In a literature review of 44 articles covering STEAM education, Perignat and Katz-Buonincontro [73] found that educators agree that STEAM enhances creativity and thinking skills. However, it is not always clear how to add arts to the mix.

Although several CT-based STEAM curricula have been suggested in the literature, research indicates that these commonly lack apparent pedagogies [91]. Hence, suitable instructional plans and supporting learning activities are needed for integrating CT in a STEAM context.

Simultaneously with the move towards STEAM, maker culture has become increasingly common in both K-12 education and informal learning settings. Maker culture has its origin in the do-it-yourself (DIY) culture. It is described as "a technology-based extension of DIY culture that intersects with hacker culture […] and revels in the creation of new devices as well as tinkering with existing ones" (Wikipedia). The popularity of "making" originates in, for instance, the idea of helping students learn how to be producers in the tech domain rather than merely being consumers of existing solutions and tools [36]. The increase in affordable and available tools needed for making activities, such as electronic kits, microcontrollers, and 3D printers, has made it easier for anyone to engage in these types of activities.

Making lets students use various materials and resources as they "embrace tinkering, or playing, in various forms of exploration, experimentation, and engagement, and foster peer interactions as well as the interests of a collective team" [98], p. 35. Making activities are interest-driven and creative, support tinkering and play, and encourage collaboration [67]. When engaging in a maker activity, students can implement their own ideas and are encouraged to let their interests and abilities guide the process [46,49], which can increase their personal investment in the task [59]. Maker activities are also seen as a way to bridge the gender gap in STEAM education [61].

While maker culture initially had a rather heavy focus on STEM subjects, there are many openings to include arts. For instance, Lindberg and others [56] studied the opportunities of arts in maker education and found that making became a means of "personal, artistic expression with quite literal layering of coded meanings." It has also been argued that arts and design practices can develop students' curiosity, and such an approach could therefore play a role in addressing the decline in interest towards science [52].

Integrating making in education can offer several benefits to students, but it also requires new skills from the teacher. As STEAM and maker culture are rather new phenomena in K-12 education, most in-service teachers have not received any training in maker pedagogies. Consequently, there is a gap between teacher education and desirable classroom practices [111] as well as a lack of understanding for how teachers are to integrate making approaches in their teaching [37]. Maker pedagogies also imply that the teacher needs to be comfortable with not knowing exactly where a lesson will end nor all the details of the equipment and materials used [50].

To support novice maker teachers, training should start with a limited set of tools and co-created lesson plans [41]. In addition, Hughes and colleagues [37] found three major themes for supporting teachers in becoming "making educators": 1) personal and professional identification with the values of the maker movement (playfulness, experimentation, problem-solving aptitude), 2) familiarity with curricular objectives and proficiency with interdisciplinary planning, and 3) support from a multidimensional maker culture for continuous development.

2.4. CT in the Finnish national curriculum

A new national curriculum for grades 1–9 (approximately ages 7–15) was accepted in Finland in 2014 and came into force gradually starting in 2016. The curriculum does not mention CT explicitly but introduces programming as part of digital competence, which includes four areas: 1) practical skills and own production, 2) responsible and safe ways of working, 3) information management as well as exploratory and creative work, 4) communication and networking. Digital competence is in turn one of seven dimensions of the so-called transversal competence, which plays a central role in the curriculum as a whole: 1) thinking and learning to learn, 2) cultural competence, interaction and self-expression, 3) self-care and managing everyday life, 4) multi-literacy, 5) digital competence, 6) working life skills and entrepreneurship, and 7) social participation, influence and building a sustainable future [28].

These seven competencies are closely related to the 21st century skills, such as critical thinking, communication, collaboration, problem-solving, creativity and digital competence [6,65]. They are not taught in isolation but are to be covered throughout the curriculum, and offer opportunities to combine different subjects, topics, and content areas. According to the Finnish curriculum, digital competence – including programming or CT – should therefore be integrated into other subjects. In addition to this implicit integration, programming is explicitly included in two subjects (mathematics and sloyd).

In mathematics, the focus is on logical thinking, problem-solving and algorithms, starting with creating and testing step-by-step instructions in grades 1–2, followed by programming in visual environments in grades 3–6. In grades 7–9, students are to deepen their programming skills and develop good programming practice. Programming in mathematics can thus be completely unplugged in grades 1–2 [5], while computers, robotics or other digital devices should be used starting in grade 3. The Finnish curriculum does not state when text-based programming languages should be introduced. The choice of programming environment is instead left to the teacher, as it depends on the learning objectives and the types of problems to be solved.

Sloyd is a subject taught in the Nordic countries and can be described as crafts, handicrafts or handiwork using both soft and hard materials. As programming became part of sloyd education, new materials and tools were introduced, such as microcontrollers, electronic components, 3D printers, and robotics. Introducing programming in sloyd is hence closely related to the maker culture discussed above. Sloyd has been part of the Finnish curriculum for over 150 years, so schools in Finland have had their own kind of maker activities long before the establishment of the maker movement [8].

Finally, in addition to viewing programming as a tool for solving problems, implementing ideas and expressing oneself creatively, it can also be introduced from a societal point of view in subjects such as religion, history and social sciences. For instance, questions related to ethics, safety and privacy can be pondered in connection with discussions on the role of algorithms, data and programmable devices in society, as well as potential future scenarios.

Introducing programming in the curriculum placed teachers in grades 1–9 in a new situation, as they were expected to teach a topic entirely new for most of them. As stated in the introduction, both international and national studies [79,86,92,106,107] have shown that teachers lack previous programming experience and struggle when trying to integrate it into their teaching. The curriculum revisions naturally required – and continue to require – time and resources, including professional development and teaching materials.

2.5. Interdisciplinary teaching

As digital competence in general, and programming in particular, is introduced as a transversal skill set, the curriculum calls for interdisciplinary teaching approaches. In addition, the Finnish curriculum defines cross-curricular learning areas that should promote students' 1) understanding of the relationship and interdependencies between different learning contents, 2) ability to combine the knowledge and skills provided by different subjects to form meaningful wholes, and 3) ability to adopt and use these in collaborative learning. Every student should participate in at least one cross-curricular learning area every school year.

Research indicates that integrated approaches can benefit both learning outcomes and personal development, for instance in terms of self-regulation, creativity and emotional health [23]. Hus and Grmek [38], p. 160] point out that "besides acquiring new knowledge, the emphasis is also on motivation, acquiring practical knowledge and developing social learning." Interdisciplinary teaching can also enrich a student's "lifelong learning habits, academic skills, and personal growth" and support the development of "advanced thinking skills leading to discovery and real-world problem solving" [45]. Moreover, integrated approaches can make fragmented or crowded curricula more efficient and coherent [23].

Naturally, there are also potential disadvantages related to integrated approaches. Assessment may be more difficult when many subjects are intertwined, and such instruction may be seen as reducing the status of a given subject [45]. Maybe the most common challenges are, nevertheless, quite practical in their nature. Interdisciplinary teaching requires planning and time, which is scarce in most teachers' workdays. It also takes a collaborative effort, which may not be easy due to both logistical issues regarding scheduling and what Drake and Reid [23] call "territorial battles" and "teacher identity".

Interdisciplinary approaches naturally require good planning for them to be implemented in practice. Different strategies have been identified when planning and carrying out interdisciplinary programming projects in grades 1–9 [108]: "1–2–3″, "spiral", and "together". The first of these, 1–2–3, refers to a situation where teachers plan the work so that it is divided into subject-specific parts, which are completed one at a time. The spiral strategy is similar but occurs in a cyclic manner, where students re-encounter the project in the same subject several times. Finally, the together approach describes a situation where students work continuously and seamlessly with the project in different subjects.

3. Method

3.1. Study setting

As the new Finnish curriculum came into force in the fall 2016, teachers faced a situation requiring training and ideas for how to introduce programming in different ways. The goal of our study was to create and evaluate a model addressing three of the objectives presented in the curriculum: teaching programming and CT in a cross-curricular manner while at the same time covering the aspects of interdisciplinary teaching put forward in the curriculum (understanding co-dependencies between different subjects, combining knowledge and skills from different subjects to meaningful wholes, and learning together with peers). In addition to covering parts of digital competence, we also wanted the model to make possible the aims of the first dimension of the transversal competence, that is, thinking and learning to learn (freely translated from the Finnish curriculum text):

> *"Thinking and learning skills lay the foundation for the development of other competencies and lifelong learning. […] This competence is promoted through exploratory and creative ways of working, collaboration, and opportunities to go deeper and focus. Teachers should encourage students to trust themselves and their views while being open to new solutions. […] Students should be given space to ask questions and be encouraged to look for answers, to listen to the views of others while at the same time reflecting on their own knowledge. […] Students should learn to use knowledge independently and together with others to solve problems, argue, reason, draw conclusions, and innovate. […] Innovative solutions presuppose students to learn to see alternatives without prejudice, combine different perspectives and use their imagination to transcend existing boundaries."*

To accomplish this, we designed a cross-curricular teaching sequence in which small groups of 3–5 middle school students were to create interactive installations of stories they wrote collaboratively. The installations were to be created using arts and design practices, while interactivity was to be achieved using tools common to the maker culture: a simple microcontroller (Makey Makey), basic electronics (LEDs, batteries, conductive materials) and a block-based programming language (Scratch). The model was designed to primarily be implemented according to the together strategy mentioned in Section 2.5; that is, the project work should proceed seamlessly, covering different subjects.

3.2. Teaching sequence

The teaching sequence was used and evaluated in four schools (S1-S4) in Finland, involving five teachers. We collaborated closely with the teachers involved, who all participated in introductory workshops. During the first workshop, they were introduced to the teaching sequence and given the freedom to develop it further and decide on the details for it to fit their respective experience, student groups and schedules. In a follow-up workshop, the teachers were trained in the tools to be used (Makey Makey, Scratch, basic electronics). Thus, we aimed to provide the teachers with the support needed according to the research discussed above (e.g., [37,41]): we gave them the backbone for a cross-curricular teaching sequence introducing a limited set of tools, while also providing training and support. The project involved five main elements:

1. Collaboratively brainstorm and write a story on a given topic
2. Divide the story into three parts and identify events in the parts that can be made interactive
3. Design and visualize the story with props, materials and tools
4. Build and program the interactive events
5. Exhibit the installation

The final element, an exhibition, was deemed relevant to bring the students' work out of their classroom. This is a way of giving the students a voice, as considering the intended audience can take an artefact "beyond the 'making' process and into the exhibition process where the piece could stand alone and be meaningful to both the designer/maker and other viewers" [56].

The teaching sequence was designed to cover 20 lessons (á 45 min). To make it easier for teachers to decide whether the project would suit their student group and if they would have time to integrate it into their schedule, a draft outlining the project was created based on the five elements above. The main objectives of each phase are described in Table 1, together with the estimated time.

The sequence covered transversal competencies (thinking and learning to learn, digital competence) as well as concrete learning objectives in six subjects: mother tongue (storytelling, communication and writing in Swedish or Finnish), mathematics (programming), arts (drawing, painting), sloyd (building, creating), science (electronics) and music (recording and editing audio files). The cross-curricular model hence brought together academic subjects with esthetic and practical subjects.

Table 1
Main objectives and timeline of the teaching sequence.

Phase	Subjects/topics	Number of lessons (20)	Content
Presenting the project	General	1	Let students know the overall plan (a collaboratively written story to be visualized and made interactive), briefly introduce the tools to be used and divide students into groups.
Presenting the maker tools	Programming, science (physics, electronics)	2	Let students explore and modify a simple project (e.g., creating an interactive piano using conductive materials, Makey and Scratch) in order to get familiar with the tools, see what can be done and discuss how the tools could be used. Introduce basic electrical concepts: current, voltage, resistance. Let students grab something from their bag or desk, reflect upon whether it conducts electricity or not and why, before testing the conductivity using Makey.
Writing the story	Mother tongue, storytelling, languages	4	Discuss collaborative writing and best practices when creating text together. Write the story, highlighting the idea of including aspects that can be made interactive using the tools used in the previous phases. Divide the story into three parts.
Planning the installation	Mother tongue, science, arts, programming	4	Discuss the basics of project planning. Create a plan including the interactivity to be added and a list of tools and materials needed.
Presentation and feedback	Mother tongue	2	Group presentations of the plans including feedback from peers and teacher(s).
Building the installation	Programming, science, arts, sloyd, music, mother tongue	6	Creative work including, for instance, drawing, painting, building out of cardboard of wood, creating conductive sections in the installation, adding LEDs, hiding batteries, connecting electronics using cables and tape, recording voice, creating music, and programming. Documentation of the work at the end of each lesson.
Exhibition	Mother tongue, digital competence	1	Exhibition for students, teachers and parents. Filming of all installations.

3.3. Pilot study

A pilot study involving 12 students and two teachers was conducted before the actual study. The project was carried out according to the drafted sequence, with 1–2 researchers observing the work in the classroom. The pilot showed that the estimated time slots for the different phases of the project were suitable. For instance, six hours proved enough for the students to build and finish the installation without the need to hurry. No changes were hence made to the timetable.

The pilot study did, however, highlight aspects of the content that needed to be revised. For instance, the pilot indicated that more practice was needed for the students to work with Scratch and build electrical circuits with the Makey Makey. The teaching material was therefore further developed by providing additional examples for the teacher to go through in class. Recording sounds in Scratch turned out to be challenging for some groups, which called for more detailed instructions on how this is done in practice (e.g., making sure the browser has access to the microphone) and the importance of finding a quiet space when recording. The pilot also highlighted the need for rotating roles within the groups so that all students could try the different parts of the project.

During the pilot, some students, for instance, chose to only focus on their favourite parts, resulting in some students not programming or working with the electronics at all. Finally, the number of LEDs per group was limited to three for the circuits to be more manageable and easier to debug. Also, as different types of LEDs may need different amounts of voltage, the decision was made to restrict the choice of LEDs so that a project only included LEDs requiring the same amount of voltage.

3.4. Data collection

The research questions were addressed based on student and teacher written diaries as well as photos and video material of students' prototypes, intermediate versions and final installations. All in all, the following data were collected: 163 student diary entries (from 71 students), six (6) stories and learning diaries (from 27 students making up six groups[2]), 14 teacher diary entries (from 5 teachers), as well as 85 photos and 15.6 min of video material documenting the process and outcomes of the students from the four schools. The results ranged from interactive two-dimensional posters to installations with spatial arrangements. The data collected are summarized in Fig. 1.

To answer the research questions, the intention was to study complex activities in the context they occurred. Such research settings advocate for collecting "messy data" [14]. Using several types of qualitative data made it possible to review different perspectives on the process and the outcomes. Furthermore, triangulation was applied in three ways: (1) data triangulation; the use of a variety of qualitative data sources, (2) investigator triangulation; the use of several different researchers in the analyses of the data, and (3) methodological triangulation; the use of multiple methods of analysis [18].

As part of the data consisted of self-reported diary entries, consistency was promoted by providing students and teachers with guiding questions for writing the entries. Students and teachers were asked to reserve five minutes for answering the questions, and the short amount of time naturally limited the number and depth of the questions asked. The questions were designed to be reflective and aimed at bringing light on the process from both a student's and a teacher's perspective. The students were given four questions, which they were to answer three times during the project: at the beginning, mid-through and the end. The teachers similarly built their diaries based on a set of questions, but these differed depending on the time they were asked (see Table 2).

In addition to the qualitative data mentioned above, two researchers visited the classrooms during the project, observing, and taking photos and videos. The pictures and videos were later used to get further insight into both the progress and the final results. Other than that, the observations mainly served as a way for the researchers to follow the work at school and help the teacher if needed.

3.5. Analysis

In the first stage of the analysis, each learning project was reviewed and described based on the students' and teachers' diaries, as well as photos and video material. The goal was to create a joint view of what type of projects the students had made, and researcher discussions were an essential part of this stage. As the student diary entries were the largest source of information over time, they were used as the primary data source. The teacher diaries, photos and videos were used as supporting data to provide more detailed insight during the analysis.

The initial review of the qualitative data and the subsequent negotiations were the basis for developing a coding scheme for the student data. The diary entries were thematically analysed using predefined categories selected and designed by four researchers based on one of the CT frameworks presented above [3] and the Finnish core curriculum. The CT framework was chosen based on two reasons: 1) it had been used previously in teacher training efforts in Finland, and 2) was deemed to capture several concepts and approaches mentioned in the Finnish curriculum. The process resulted in six CT-related categories aimed at revealing to what extent students were involved in activities dealing with CT concepts or approaches. The categories are similar to those used in the framework but based on the data review, some related categories were combined to make it easier to map excerpts from the diary entries to a given category. In addition, three of the original concepts were not found during the initial review (algorithms, decomposition, patterns) and were hence not included as categories in the coding scheme. Table 3 gives an overview of the final categories, the corresponding original CT concepts and approaches, as well as the connections to the curriculum.

In addition to the six main categories, two categories were included to capture students' feelings of 1) success and joy as well as 2) frustration and challenges, as such emotions may affect students and their learning in different ways [70]. Since these emotions can be affected by classroom factors, such as instruction and curriculum content [71], the added categories were considered important to get insight into how students experienced the project and consequently be able to draw potential conclusions about the cross-curricular model. The resulting coding scheme is shown in Table 4.

When the coding scheme had been designed, the student diary data were coded using content analysis; that is, the textual material was analysed, reduced and summarized according to, in our case, predefined themes [15]. Four researchers participated in the analysis phase. The student data were analysed first, and the findings were shared, compared, discussed, and refined in analysis meetings. Relevant excerpts from the student diary entries were selected to be included in the article to exemplify the categories. Finally, a frequency analysis was conducted of the six CT categories, resulting in an overall view of what categories were mentioned the most: what CT activities students seemed to engage in the most, and what activities were less common.

Next, the teacher diary data were analysed following a similar process to the one described above, except for the final part (frequency analysis). The goal was to get insight into the teachers' experiences working with the teaching sequence at the beginning, mid-through and at the end. The teacher data coding scheme was finalized after the student data had been analysed and coded for it to include any potential additional areas arising from the student data. The final coding scheme included five main categories – experience, expectations, implementation, challenges, and surprises – as presented in Table 5.

The coding phases are visually depicted in Fig. 2.

The results from the coding phases and the photos and video material were used to bring further light on RQ1 – *How can a cross-curricular project model including CT be designed for middle school level?* While the question has already partly been addressed in the presentation of the teaching sequence in Section 3.2 above, that discussion covered the intended model. However, it is equally important to also address the actual implementation, e.g., How many hours were used and how many school subjects were involved in practice? Was the final project based on a story, and how well did the installation support it if so? How esthetic was the final installation (i.e., including different materials, hiding cables, etc.)? The implementation process was evaluated using these questions based on the diary entries, photos and video material.

3.6. Ethical considerations

As the students in the participating schools were minors, parental consent was needed. All parents were given information about the project, its objectives, and the data to be collected, after which they were asked to approve or disapprove their child's participation in the study. Next, the teachers in each school created a list of aliases for the students

[2] The stories and learning diaries by the 27 students were from school S4, which carried out the project according to our original plan. Both the stories and the diaries were created groupwise, which has been taken into account during the analysis as well as the presentation of the results below.

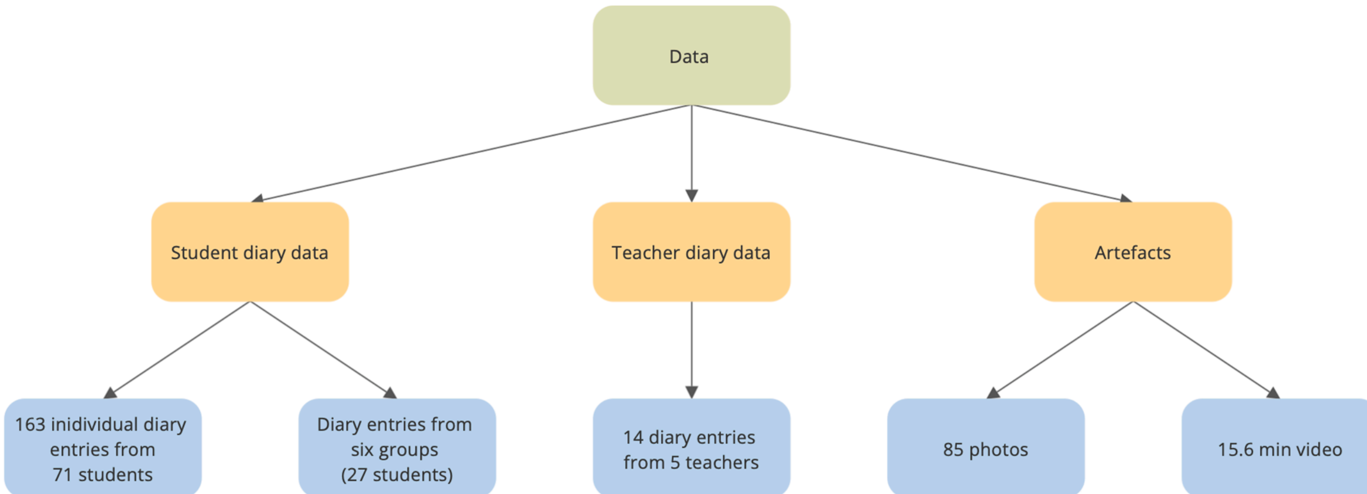

Fig. 1. Collected data.

Table 2
Questions to students and teachers at the beginning, middle and end of the project.

	Beginning	Middle	End
Students (n = 98)	(1) What did you do? (2) Tell us about two things, in which you succeeded. (3) Was something difficult or troublesome? If yes, tell us about those situations. (4) What was the best?		
Teachers (n = 5)	(1) Describe what kind of experiences you have with programming/ circuits/xxx? (other related issues in the project? (2) What expectations/ hopes/aims do you have for the project from your students' perspective? (3) What expectations/ hopes/ aims do you have for the project from your own perspective? (4) (4) How easy/ difficult do you think it will be to adapt the teaching sequence? Do you think you will modify it to suit your students, or have you already modified it? If so, how?	(1) What was it like to implement the project from a teacher's perspective? (easy/difficult/ challenging) (2) How have the students come along/engaged in the project (challenges/ successes)? (3) How well did the teaching sequence work? Have you made any adaptations to it during the project?	(1) What was the project like from a teacher's perspective? (easy/ difficult/ challenging) (2) How did the expectations/ hopes/aims you had for yourself and your students realize? (3) How well did the teaching sequence work? (4) How do you feel about the outcomes of the projects? (5) Where could you use this kind of type of activity, in which subjects? How could you use it in the future?

Table 3
The categories used in the coding scheme, the corresponding CT concepts/approaches and their connection to the curriculum.

Category	Corresponding CT concept/approach	Connection to the curriculum
Step-by-step instructions and algorithms	Algorithms	Step-by-steps instructions is the term used to introduce algorithms
Reason, consequence and causality	Logic	When solving problems and innovating students should argue, reason and draw conclusions.
Creating, tinkering and debugging	Creating, tinkering, debugging	Student should, among other related learning objectives, engage in practical production, as well as exploratory and creative ways of working.
Collaborating	Collaborating	Collaboration is mentioned throughout the curriculum.
Abstraction and conceptualization	Abstraction	In addition to abstractions and generalizations, students need to learn terminology that aid in communication and collaboration.
Persevering	Persevering	Persevering is key for developing both the competences in the curriculum and lifelong learning.
Evaluation	Evaluation	Students are encouraged to ask questions and look for answers, to listen to the views of others while at the same time reflecting on their own knowledge.

who were allowed to participate in the study. The aliases revealed only the gender and the school of each student. Throughout the study, the researchers only saw and used the aliases and never had access to the students' names or any other personal information. Moreover, all photos and video material were captured so that potential students close by were not recognizable.

4. Results

In this section, the results are presented, focusing on 1) the implementation of the teaching sequence and 2) the categories used in the students' coding scheme. When needed, examples are provided in the form of diary excerpts and photos.

4.1. Implementation

At the onset, it was clear that the schools would not implement the teaching sequence in the same way. While S4 aimed at completing the project according to the original plan (20 lessons including several

Table 4
Coding scheme used for student diary data.

Student	Reason, consequence and causality			Creating, tinkering and debugging			Collaborating			Abstraction and conceptualization			Perseverance			Evaluation			Success and joy			Challenges		
	Start	Mid	End	Start	Mid	End	Start	Mid	End	Start	Mid	End	Start	Mid	End	Start	Mid	End	Start	Mid	End	Start	Mid	End
G1																								
B2																								
…																								

subjects), schools S1-S3 opted for completing it on a tighter and more intense schedule resulting in longer sessions at a time over a few days, rather than spread over several weeks. Although the total number of hours was not much lower than in S4, the tighter schedule greatly impacted the process and the results. As the teachers could not skip the introduction (presenting the project and the tools) nor the creation (planning and building) parts of the project, most teachers in S1-S3 chose not to require their students to write a story, present their plans or give feedback on the other groups' plans. Hence, the learning objectives of storytelling, collaborative writing, presentation, communication, and feedback were left out. This also meant that the role of the mother tongue subject in the project notably decreased.

Without a story, the final products became smaller and more "ad hoc creations" rather than complete installations. At the same time, the connection between the subjects suffered due to a lack of time. This does naturally not imply that the products were of poorer quality, only that these projects did not fulfill the original plan of transforming a story into an interactive piece of art. Fig. 3 shows two examples of artifacts that were not built based on a story. The example on the left is an interactive radio made from a cardboard box. The user can press the "buttons" and hear different sounds. On the right is an interactive poster, providing information on a given topic both in writing and by touching the conductive parts of the installation.

In school S4, all groups created interactive story installations, starting from collaboratively writing a story and dividing it into three parts, creating a plan for how to visualize the story and make it interactive, presenting the story and the plan, and making adjustments before starting the building phase. They ended up using more than 20 lessons, needing more time for the introduction phases to ensure that the students had sufficient background and skills to get started with the project. The process is shown in Figs. 4-6, beginning with a group of students reviewing their story and the corresponding plan in Fig. 4.

Fig. 5 shows a ready-made installation. The three parts of the story are visible as three connected backdrops. The story is written on the white notes and prompts the user when to push, touch or otherwise interact with parts of the installation.

As aesthetics was highlighted throughout the project, students also spent time hiding cables and making the final product as "clean" as possible. This is seen both in the interactive radio (Fig. 3) and the three-part installation (Fig. 6).

Time was the main challenge when implementing the project, particularly in schools S1-S3. The teachers, for instance, mentioned extra events and holidays resulting in insufficient time for the project after all compulsory schoolwork had been taken care of[3].

> *Due to coincidences, the schedule turned out to be too tight for our group. In addition to the winter holiday, we have a school trip and two thematic days. Without these, we would have had time to internalize things better, and the final results might have been "deeper". Now some [students] finished the project hastily due to the plans changing too thoroughly (S1).*

The lack of prior knowledge was also problematic, mainly from a teacher perspective. Taken together, the lack of time and experience brought initial feelings of uncertainty, here described by the teachers in S1 and S4.

> *The project has been challenging for me. I did not have time to absorb things as thoroughly as I should have. (S1)*

> *In the beginning, it was kind of difficult to know how everything would turn out; both I and the students were curious but still uncertain. […] Their background in programming and the way of thinking it requires*

[3] All excerpts have been translated from Finnish or Swedish by the authors. Underlining is used to show the reason for why a given excerpt belongs to the corresponding category.

Table 5
Coding scheme used for teacher diary data.

School teacher	Experience			Expectations			Implementation			Challenges			Surprises		
	Start	Mid	End	Start	Mid	End	Start	Mid	End	Start	Mid	End	Start	Mid	End
S1															
S2															
…															

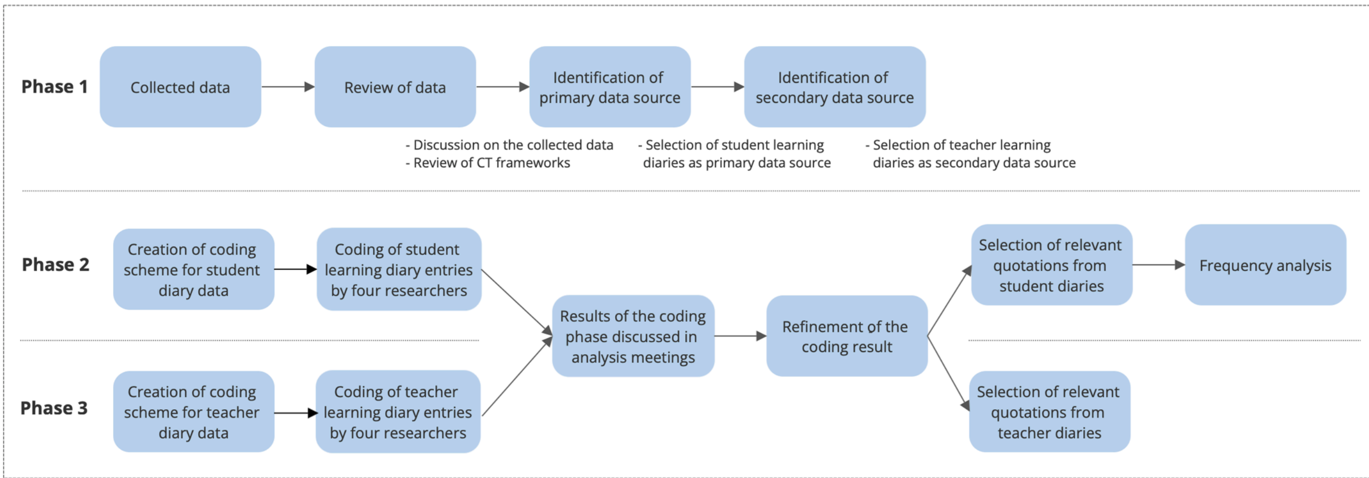

Fig. 2. The three coding phases.

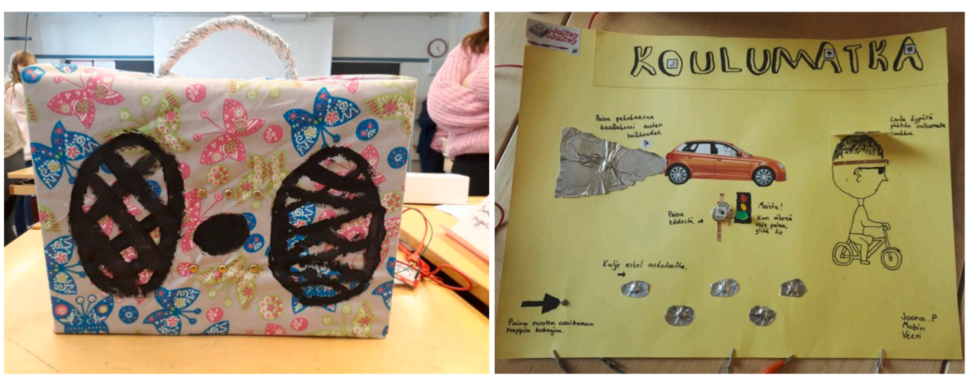

Fig. 3. Smaller-scale projects not based on a story.

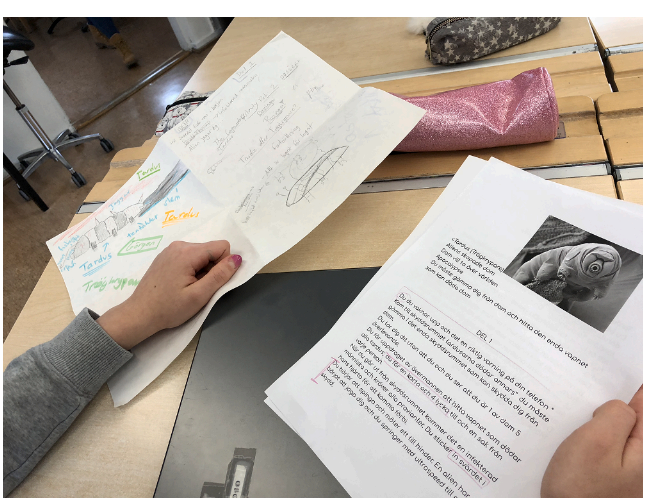

Fig. 4. A student group reviewing their story and the corresponding plan.

differed. I felt that I was not enough; getting lots of questions and being alone with 21 students was not optimal. (S4)

The teachers noted that some of these challenges were eased by the student-centered and collaborative approach as the students took responsibility for their projects and managed to work in groups. In general, the negative feelings were relieved as the work progressed:

I managed to get the students excited, and many other children have come by our classroom to spy on what we are doing. (S1)

As we got to the planning and building phase, everything went well. I think it was easier to understand when it became more concrete. […] Then I almost felt like I was redundant. (S4)

After completing the project, the teacher was positively surprised by the students' work and their final installations while seeing opportunities to use the same project model further on.

Some of the results were very high quality […] the journey became more important than the final artefact. For my creative students, this [way of

Fig. 5. A finished installation based on a story in three parts.

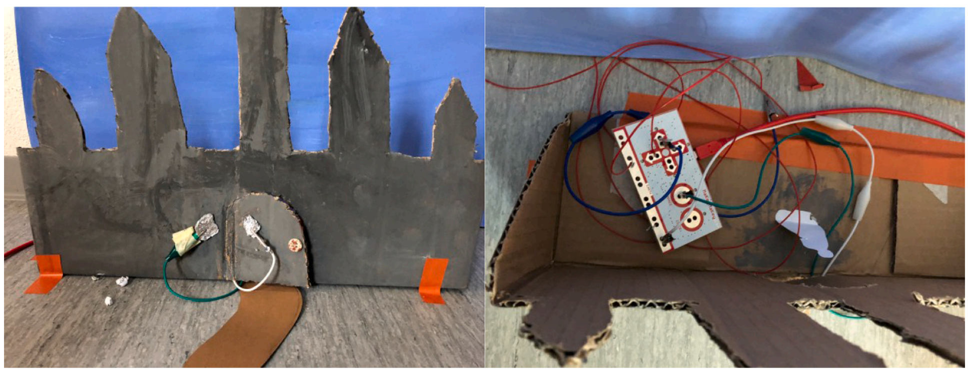

Fig. 6. Cables and electronics hidden in the installation.

working] will give rise to something more still this spring. We have only just begun. (S1)

My expectations were exceeded, as all groups ended up with something that worked. I think this model could be used in any subject, and we are already planning to create a story path along which other student groups can listen to different stories. (S2)

I am so pleased, and it was nice to see how the children's ideas changed and visions grew. […] This kind of project could be used in any subject. It comes down to your own ideas and daring to try. I will definitely use this way of working in the future. (S4)

4.2. Diary entries

Table 6 gives an overview of how frequent the different categories were in the students' diary entries: the number of explicit comments relating to the category as well as the number of individual students and student groups mentioning the category.

Table 6
Overview of the category mentions in student diary entries.

Category	# mentions	# students (out of 75)	# groups (out of 6)
Reason, consequence and causality	22	17	0
Creating, tinkering and debugging	200	61	6
Collaborating	55	3	1
Abstraction and conceptualization	52	22	6
Persevering	18	10	4
Evaluation	56	28	3
Success and joy	189	67	6
Frustration and challenges	122	51	6

The second category – creating, tinkering and debugging – was the most common one, while the fewest occurrences were found for the categories "reason, consequence and causality" and "perseverance". Most students also discussed success and challenges in their diaries. The frequencies shown in the table only account for students explicitly mentioning the given categories in their diaries. In addition, some of the categories could also be implicitly seen in the students' answers, for instance, students writing in "we" form, indicating collaboration taking place. In the following, each category is reviewed separately.

4.2.1. Reason, consequence and causality

Most comments falling into the first category, *Reason, consequence, causality* (22 mentions by 17 students and 0/6 groups), were related to the students' project, how it worked and what it did (Table 7).

Both excerpts in the table highlight how the parts of the final installations were connected, as well as the cause and effects when interacting with them. The second excerpt relates to the interactive radio in Fig. 3.

The teachers did not mention this category in their learning diaries. However, due to the nature of the projects, the work naturally called for reasoning and discussions on causality, actions and interactions when making design decisions, e.g., what should be done to arrive at a given result, and what the consequences would be if choosing a given design

Table 7
Diary excerpts related to reason, consequence and causality.

Reason, consequence and causality (22,17,0)	We made a teddy bear. It had a fork and knife as hands. And when these were pressed, a voice said what the thing was. We made a radio with six buttons out of a cardboard box. There was a handle made out of folio on top of the box. When you pressed a button you had to hold the handle at the same time. The buttons made different sounds.

path? As such, the task of building an interactive piece partly in itself raised the need for the reasoning and consequence aspects.

4.2.2. Creating, tinkering and debugging

As already discussed above, the scope of the project varied between schools. While the teacher in S4 followed the original plan, students in the other schools worked on smaller projects. Regardless of the scope or nature of the project, they all involved *creating, tinkering and debugging* (category 2, 200 mentions by 61/75 students and all six student groups) as students mentioned concrete activities such as programming, recording sounds, connecting cables, drawing pictures, writing stories, and so on. Table 8 presents the main types of comments related to this category.

Some students described their projects in detail, highlighting the connection between design and functionality. In addition, as all projects were to be interactive, there was a need for tinkering, exploration and – if things did not work – debugging. Most problems that called for correcting an error or re-exploring the situation were related to circuits or programming. Finally, students mentioned looking for creative solutions and learning by exploration and experimentation. This was also highlighted by the teachers, who were positive about both their students and themselves exploring and experiencing new areas. In school S4, where the interactive installations were relatively large and demanding, the teacher guided the students to explore several solutions when implementing their designs. Tinkering and debugging were natural parts of the process, highlighted and encouraged by the teacher:

> *"In the beginning, everything was new, and it was perhaps hard to know how to structure everything to arrive at a good outcome. The trickiest part was to get the students to understand how to think when they were to start planning and building the concrete parts. [...] Slowly, they built a picture of what they wanted. Much has changed on the way. The challenges have varied [but the] students have dared to try, helped each other and found out [on their own]." (S4)*

All in all, 74 students stated that they had been engaged in creating, exploring and debugging activities while completing the project. In S1, S2 and S3, where the students worked on smaller interactive projects, the tinkering became more a matter of fact without teachers explicitly explaining its importance in the project.

4.2.3. Collaboration

Students from all schools mentioned *collaboration* (category 3, 55 mentions by 33/75 students and 1/6 groups) and working in teams in a positive manner, either explicitly or implicitly (Table 9). They appreciated working with others and acknowledged the importance of the other team members in finishing the project. They managed to divide the work among the team, found their own roles and encouraged each other. The importance of collaboration can also be found implicitly in several other comments, where students write in "we" form. While most students considered collaboration both an essential and a positive aspect of the project, some students also found it difficult and more stressful than individual work.

Table 8
Diary excerpts related to creating, tinkering and debugging.

Creating, tinkering and debugging (200, 61, 6)	The project was a box, which we painted pink and wrote "party" and "magic box" on. We put an old school tray on the box, and on it a folio cup with marshmallows on metal sticks, three plates covered with folio. One plate had a cloth donut which had iron thread in it and decorations that conducted electricity, one had gingerbread dough on it and the third one was empty. And then there was a cup which was also covered with folio. The folio cup repeated the sound "drink", two plates made drum sounds and the donut said, "don't eat me" and the side of the box played music. It was difficult to get the cables in the right places. Every now and then the cables did not work. To get the sound to play from different things; sometimes the cables were connected incorrectly and there was no sound. I programmed and explored a lot of new things. The best thing was to invent.

Table 9
Diary excerpts related to collaboration.

Collaborating (55, 33, 1)	The best thing with the project was that I could be with my friends and plan it together. The best thing was to do the project when we had a good team to do it with. I did well on everything, not just me but the entire team. We managed to divide the work and collaborate. We found a box and then A got a great idea to build everything around the box. While A and B built the box, C made a city and D programmed a maze. Then we connected the cables, added sound effects, and fixed the last things. And then it was done. We encouraged each other. I managed to come up with an idea for our work and distribute roles. The teamwork and stress were most difficult.

Fig. 7 shows the box project described in the second excerpt of the "division of labor" subcategory. This project included more programming than the others, as one of the students took on the role of software developer and designed a maze game to be included in the final project. In addition to the box having interactive elements, it also included a laptop with a game to be finished to complete the storytelling.

The teacher diaries indicate that they expected teamwork to be part of the project through comments such as "*students will learn from each other*". This is no surprise, as collaboration was a central part of the teaching sequence, and the projects were to be done in groups.

4.2.4. Abstraction and conceptualization

Students' descriptions of what they did and how things worked highlighted their use of scientific terminology, with some *abstraction and conceptualization* (category 4, 52 mentions by 22/75 students and 6/6 student groups). Quite a few students used concepts and structures that presented abstraction capabilities. All in all, 52 conceptualizations were found in the students' comments. Some students described their work on electronics with everyday terminology, while others used more exact terminology, such as circuits, cables, and switches (Table 10).

Most of the comments with exact electronics terminology came from students in schools S3 and S4. While one might expect that the teachers in these schools were confident with electronics and electrical circuits before the project, this was not the case. In fact, these were the two teachers who reported to be the least confident with this topic in their first diary entry. For instance, the teacher in S3 stated that electrical circuits were familiar mainly from her time as a high school student.

In schools S1 and S2, the teachers seemed more worried about the schedule and how to motivate all students. Although the project was not primarily about electronics, the teachers in these schools made a connection from the project brief to that specific curriculum topic. For instance, the teacher in S1 had high expectations on the project in terms of promoting students' understanding of electronics, conductivity, and the functionality of specific components, such as conductors and insulators.

4.2.5. Perseverance

When debugging and trying to learn something new, *perseverance* (category 5, 18 mentions by 10/75 students and 4/6 groups) is often key. The student diary entries (Table 11) showed how students overcame obstacles and challenges by collaborating and asking for help from their peers and teachers. Other essential aspects of persevering were related to recovery from failure and not giving up. Even though things

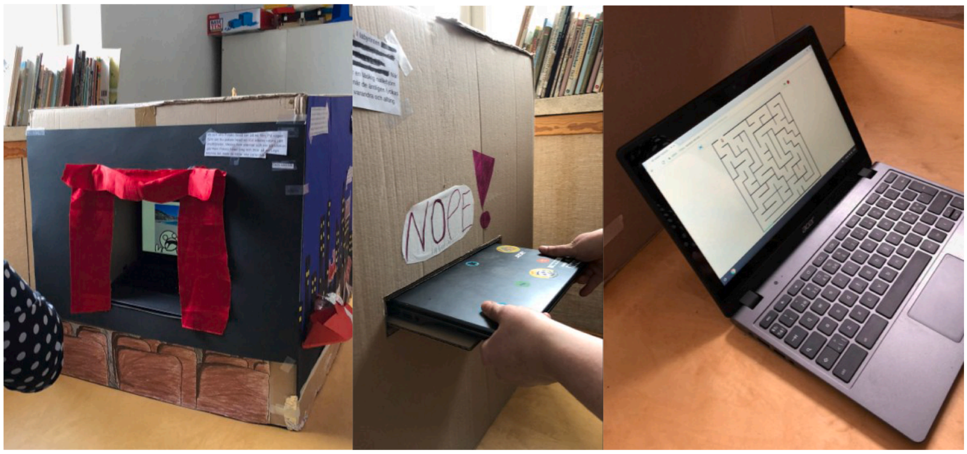

Fig. 7. A story visualized and made interactive using a cardboard box.

Table 10
Diary excerpts related to abstraction and conceptualization.

Abstraction and conceptualization (52, 22, 6)	When you hold the folio ball and touch a thing that has a cable connected to it, it makes some noise.
	Connecting the threads, there were so many threads.
	I made an electric switch; I managed to come up with a way to hinder the flow of electricity.
	We got to learn about electrical circuits, circuit diagram and the symbols that should be part of a circuit diagram.

Table 11
Diary excerpts related to perseverance.

Persevering(18, 10, 4)	At first we didn't understand what cable to put where, but then we succeeded.
	After many "buts", our story was finally written.
	The Makey was fun to work with, but when we saw it for the first time we thought "No, we'll never manage this". The Makey looked really difficult, but it turned out to be much easier than what it looked like. It took quite some time for us to come up with how to program it, but then we came up with it.
	We were almost done but then the Makey fell to the floor. And we had to do some things all over again.
	It was hard when the circuit thing did not work, but luckily in the end it started to work.
	Then we programmed. We did that for quite some time. But then we finished.

did not go as planned, students did not give up but tried again until they succeeded.

The teacher's role in supporting and maintaining students' perseverance varied. In S4, the teacher was coaching and encouraging students throughout the 20-hour process, while in the other schools, students had a much more limited time to design and implement their idea. Consequently, the respective teachers also had less time and opportunity to encourage students' perseverance, as the deadline was coming up quite quickly. The project also required perseverance from the teacher, as many of the topics were new to most of them. For instance, at the beginning of the project, the teacher in S4 noted, "*I will have to dare, dare to do something that I might not yet be 100% sure of, but I will also learn on the way*".

4.2.6. Evaluation

Students *evaluated* (category 6, 56 mentions by 28/65 students and 3/6 student groups) their projects from different perspectives

Table 12
Diary excerpts related to evaluation.

Evaluation (56, 28, 3)	I think our project was too simple compared to the others.
	The final result was very good and not too challenging to implement.
	Our project was simple, but still many came to see it and we received positive feedback.
	Our project was successful, and I'd say it would get a 9.
	I think our project succeeded well, but due to some difficulties
	I would give our work the grade 8.5 out of 10.
	It worked well and was beautiful.
	It succeeded according to our plans. It was fun.
	We planned to have it on three walls, but it became a box, so it turned out completely different from what we had thought. But also, much cooler.

(Table 12). Some focused on the experienced difficulty level, while others took a more school-like approach and gave the project a final grade (on the scale 4–10, as commonly used in the Finnish school system). Some students focused on the aesthetics, and others reviewed their final product in relation to the original plan and vision.

The teachers also reflected on the students' projects and learning to evaluate their own original expectations against what really happened. The teachers pointed out that the students had learned both subject knowledge and meta-skills (such as exploration, planning and collaboration) while seemingly enjoying the process and showing ownership of their work. For instance, in S4, the teacher stated that

> *"[t]he students have learned a lot during the project, and I could not have imagined the result. Everything turned out so much cooler and better than I thought. The goal of having students feel comfortable with this type of project was met and their work is impressive. … Most importantly, the students are pleased and, above all, proud of their work." (S4)*

4.2.7. Joy and challenges

Interestingly, no student or student group found their project too difficult. These feelings can be recognized from the students' comments that were coded into the two final categories: students' experienced level of *success and joy* and *frustration and challenges* (Table 13).

Almost all students (67/75) and all six student groups mentioned aspects of being successful and being happy. Similarly, most students (51/75) and all student groups also reported on different types of challenges during the process. Although students mentioned times of frustration while working on the project, they simultaneously experienced success and joy. With the challenges not being too difficult, overcoming these naturally resulted in positive feelings.

Table 13
Diary excerpts related to success and joy, and frustration and challenges.

Success and joy (189, 67, 6)	It was very fun to work with the cables. Our final result was a bit different than we planned, but we were still happy. The best thing was to succeed. It was really fun! We succeeded really well and we probably did the best in placing the cables. The best part was to figure out how electricity flows. The best thing was to finish the project.
Frustration and challenges (122, 51, 6)	[W]e could not find a place to record, that was difficult. It was hard to concentrate. It was hard to get the voice repeating. [W]e nearly ran out of time. We crawled forward.

5. Discussion

In this section, the results presented above are discussed and synthesized based on the three research questions listed in the introduction.

5.1. Cross-curricular model including CT for middle school level

The first research question dealt with the cross-curricular teaching sequence. Curriculum reforms always require some level of effort from the teachers, and the introduction of programming and digital competence as an interdisciplinary trait in the Finnish curriculum affected all teachers. Due to the nature of education in grades 1–6, most teachers are used to interdisciplinary work; most of them are so-called class teachers and have the same student group in (almost) all subjects. This makes it possible to mix and relate subjects quite easily. Nevertheless, most teachers lacked previous experience in programming or CT and hence needed to learn both the new content and how to integrate it into their teaching. As pointed out by Strickland and others [112], integrating computing topics into other subjects "is a non-trivial task that may unfairly burden elementary teachers, who are often generalists" (p. 1149).

The goal was to create a simple yet detailed project model that teachers could follow even without a background in electronics or programming. The project resulted in concrete installations ranging from interactive posters to interactive boxes, walls and games. The implementation process and the scope of the final project were, not surprisingly, related to the time spent on the project. The original plan was for the project to take 20 h, but only in one school could the teacher find this amount of time in the schedule. In her classroom, the model created a coherent whole in a crowded curriculum [23] combining many subjects and topics. All other teachers opted for completing the project in a shorter time, following a more intensive schedule. This led to fewer subjects being covered and fewer cross-curricular connections made. The final artefacts produced by the students were consequently also smaller.

Nevertheless, giving each teacher the freedom to adapt the teaching sequence to the local context was crucial, as differing resources and institutional support affect the everyday classroom practice. Flexibility is also important to allow for modifications on-the-fly as practical problems may arise from, e.g., holidays mixing with the schedule, having to switch rooms between sessions or students being absent. Had the model been too normative or restrictive, those teachers and students had not been able to carry out the project at all. Although the process and the final result were not as "complete" as intended, the students nevertheless were given the opportunity to experience this way of working, learning how to use the tools and completing an installation with their group.

Arts and programming both became a "fulcrum through which wider domain learning and creativity is promoted" [34], p. 167. While CT is commonly considered to be developed through programming activities [25,39,85,88], the teaching sequence seems to have fostered CT abilities throughout the project as a whole. Depending on the scope of the project, programming served as a tool supporting learning in other subjects or as a glue between several subject areas [108].

The hands-on workshop and joint planning sessions at the schools were essential to help each teacher find a suitable implementation plan as well as start and finish the project. Depending on the students' previous experience in programming, electronics, microcontrollers, and maker activities, the two lessons reserved for introducing the tools may not be enough. On the other hand, the story can be written in less time than the initially reserved four lessons, as it can be further improved in the planning phase. Based on the teachers' experiences, we have revised the teaching sequence by dividing the tool introduction into three separate phases, reserving two lessons for each. As a result, the total number of lessons has increased from 20 to 23. Some of the phases are, however, optional. For instance, there is no need to go through the basics of programming and Scratch if these are already familiar to the students. The revised plan is presented in Table 14.

5.2. Teachers' approach and experience

The teacher's role in the project varied. As already discussed in the previous section and according to Finnish practice, the teachers were free to modify the cross-curricular model to fit their student group and style of teaching. The teachers' view of the project – that is, what they chose to focus on – and the time available for completing the project naturally affected what the students did and to what extent. This was also reflected in what CT areas students acknowledged the most. For instance, some teachers' inclination to relate the project to the curriculum topic "electronics" led them to guide their students to focus on abstraction and conceptualization. This may, at the same time, have had a negative impact on aspects such as creativity, tinkering and debugging. It seems plausible that as teachers become more familiar with this type of teaching as well as the topics and tools involved, they will more easily see the model as an entity where all parts and subjects play a vital role and have their own learning objectives.

There are natural connections between CT and other subjects. Still, for teachers to be able to map CT elements to curricular content, they first need to learn to identify opportunities where and see examples of how this can be done [22,77]. There is, hence, a need for explicitly supporting pre-service [101] and in-service teachers [81] in integrating CT in the curriculum. For in-service teachers, the introduction also needs to be made to meet their goals and practices [81], which was an important design criterion for the teaching sequence presented in this paper. As such, our study has, similar to other research (e.g., [100]), highlighted the importance of and possibilities for teachers and researchers to serve as co-designers of teaching sequences and syllabi when integrating CT in the K-12 classroom.

Both student and teacher diary entries were mainly positive, indicating that they appreciated this way of working despite some practical challenges and difficulties. Compared to the positive aspects, the issues causing frustration can be considered relatively minor, as they were mainly related to project planning and organization. Similar challenges, related to, e.g., time constraints and shifting classrooms, have been found in other studies [37]. Nevertheless, such problems are everyday challenges for teachers and should not be underestimated. Organizing these types of activities in the classroom means extra work for the teachers. If they are not supported and the projects are not considered in relation to the entire school's operational framework and schedule, the risk of failing increases.

The teachers' role can also be related to their mindset and self-experienced ability to learn and develop as teachers in the project. All teachers had limited, if any, prior experience working with programming or making in their classroom, and naturally, the teacher with more time also had more opportunities to learn. Time was, however, not the only aspect but also the teacher's aptitude for this type of work, or what Hughes and colleagues [37] call "personal identification with the values

Table 14
Revised teaching sequence.

Phase	Subjects/topics	Number of lessons (23)	Content
Presenting the project	General	1	Let students know the overall plan (a collaboratively written story to be visualized and made interactive), briefly introduce the tools to be used and divide students into groups.
Introduction to programming and Scratch (optional)	Programming	2	Review of the basics of programming and the Scratch environment (if needed).
Introduction to basic electronics and Makey(optional)	Science (physics, electronics)	2	Review basic electrical concepts: current, voltage, resistance, circuits (if needed). Let students grab something from their bag or desk, reflect upon whether it conducts electricity or not and why, before testing the conductivity using Makey.
Combining programming and electronics (Scratch, Makey and electronic components)	Science (physics, electronics), programming	2	Let students explore and modify a simple project (e.g., creating an interactive piano using conductive materials, Makey and Scratch) in order to get familiar with the tools, see what can be done and discuss how the tools could be used.
Writing the story	Mother tongue, storytelling, languages	3	Discuss collaborative writing and best practices when creating text together. Write the story, highlighting the idea of including aspects that can be made interactive using the tools used in the previous phases. Divide the story into three parts.
Planning the installation	Mother tongue, science, arts, programming	4	Discuss the basics of project planning. Create a plan including the interactivity to be added and a list of tools and materials needed.
Presentation and feedback	Mother tongue	2	Group presentations of the plans including feedback from peers and teacher(s).
Building the installation	Programming, science, arts, sloyd, music, mother tongue	6	Creative work including, for instance, drawing, painting, building out of cardboard of wood, creating conductive sections in the installation, adding LEDs, hiding batteries, connecting electronics using cables and tape, recording voice, creating music, and programming. Documentation of the work at the end of each lesson.
Exhibition	Mother tongue, digital competence	1	Exhibition for students, teachers and parents. Filming of all installations.

of the maker movement". While the approach was a good fit for the teacher in S4, the ambition level was lower in the other schools. Nevertheless, with support from the researchers and the teaching sequence they were allowed to modify, these teachers could also take on the role of beginning "making educators" [37]. The challenge now is to help them continue this journey to the extent that meets the curricular requirements. As knowledge and beliefs affect teachers' practice, professional development aiming to change their practice must also address their beliefs [11].

OECD has published several position papers on the future of education, discussing the competencies, skills and values needed when educating children and youth for a rapidly changing world (e.g., [66]). One of the key aspects is to help students become "future-ready" and develop agency, that is, feel a responsibility to *"participate in the world and […] influence people, events and circumstances for the better"* (p. 4). In this context, co-agency becomes an important factor: teachers need not only focus on individual students but also recognize other, mutually supportive, relationships affecting the students' learning (e.g., with peers, families, teachers, and broader communities). The OECD concludes that *"everyone should be considered a learner, not only students but also teachers, school managers, parents and communities"* (p. 4). This co-agency could also be noted in our study, as not only individual students but also the student groups and their teachers had to develop their skills and knowledge together. School leaders and other actors that can provide resources and organizational support should be included in the co-agency to further facilitate this type of cross-curricular projects.

5.3. Students' involvement in aspects of CT

The coding scheme for analysing the student diary data was constructed based on an initial data review identifying CT concepts and approaches. This led to a scheme containing six CT aspects, which did not include three common elements: algorithms, decomposition and pattern recognition. The two latter ones may be too abstract for students to verbalize in their diary data. Students not mentioning algorithms was, on the other hand, quite surprising. Nevertheless, algorithms, or step-by-step instructions, are the foundation for programming activities in general and according to the Finnish curriculum. However, when reviewing the final artefacts and reading the diary data in detail, it became clear that programming had played a relatively small role in most student projects.

There are several potential reasons for this. Maybe the most plausible one is that most groups had used programming to add sound or voice to their interactive artifacts, which can be done with only a small number of blocks in Scratch (events and sound). One group did create a maze program in Scratch, but they talked about the resulting game, not the process of constructing it, i.e., programming. Another possible reason is that students viewed programming as a tool for accomplishing a given project task and did not distinguish between the tools used when describing their work. This implies that the model is not necessarily optimal for learning to program per se, but rather provides a creative and interdisciplinary context in which programming can be used as a tool in different ways depending on the students' skill and ambition level. Introducing CT in education as part of other school subjects, without sufficient focus on teaching the basics of programming, might hence not solve the general call for a broader understanding of computer science in K-12 education [26].

The results show that the model fostered several CT-abilities, in particular creating, tinkering and debugging. This can be seen as a natural result of the making aspect of the model, as maker activities commonly let students explore, experiment, and play together with their peers [67,98]. Similar results have been found in other studies of physical computing: the focus of the projects will affect the abilities that students develop and use [47]. Moreover, creating, tinkering, and debugging come naturally in art and design projects when the task is open enough to allow for the students' own creativity and design.

When students are asked to design and build their creative art or design piece, the assignment needs to simultaneously be (1) open enough to give space to playfulness and creativity while also being (2) sufficiently framed so that one can fail and then fix things. The nature of the project hence also highlighted the need for teaching and learning topics not explicitly required in the curriculum, such as debugging. Mistakes are made in open and creative projects and debugging plays an important role when programming and engaging in maker activities. Learning how to deal with errors is challenging, yet novices are often expected to develop debugging practices on their own [63], although studies indicate that even young children can learn debugging given age-appropriate tasks and activities [99]. When introducing programming, teachers, therefore, need to be able to help their students debug their programs. Such explicit instruction can improve both self-efficacy and debugging performance [63]. Even though students found errors and problems frustrating, the diary entries also show the joy and positive feelings of success when managing to fix the mistakes. This may partly be related to working in groups, as research has shown benefits from participatory debugging, that is, collaborative troubleshooting [87].

The teaching sequence simulated a collaborative design activity common among professional designers and artistic collectives. In the sequence, the importance of individual contributions was deliberately faded: the projects were team projects with collective responsibility. This highlights the crucial shift from the traditional view of programming as an individualistic and tool-oriented problem-solving approach towards "computational participation" [48]. Our results suggest that this was the case in all schools, as all students were able to create their artifact collaboratively. Many students were also proud of their work, indicating that they had invested personally in the task, which is common to making activities [59].

The results show that the challenges faced were real but not insurmountable. Clearly, during a short project, the deadline comes up quite fast, and there is only limited time for reviewing different ideas and solutions. This was the case in three schools, whereas the students in S4, where the model was carried out according to the original plan, had greater opportunity and more time to experience the importance of persistence in practice. The results hence indicate, in no way surprising, that creating, tinkering, and debugging require persistence over a longer period.

5.4. Limitations

The results presented in this paper are based on one case study carried out in a Finnish setting, positioning it in a specific educational context with its own opportunities and constraints. More research is thus needed for making generalizable claims. Nevertheless, as programming, CT, maker culture and STEAM are considered important areas in K-12 education at an international level, the project model and experiences described can be interesting also when considering other educational systems.

Naturally, many factors play a role in classroom settings. In this study, for instance, student and teacher demographics have not been considered. Although the student and teacher diaries provide some insight into their previous experience, attitudes, background knowledge, and school culture, no detailed information was collected about these aspects. As such, it is not possible to fully conclude what factors affected the success of the projects, and for the findings to be used as the basis for theory building, additional empirical research is needed.

6. Conclusions

In this paper, we have described the design and evaluation of a cross-curricular teaching sequence in a middle school context. The goal was to create a model to help teachers address many parts highlighted in the national curriculum, including digital competence, programming, student collaboration and cross-curricular learning. The results from our study in four Finnish schools indicate that teachers managed to integrate the model into their teaching, while the ways in which this was done differed due to the time available and the level of organizational and collegial support available. Teachers and students learned together, and our results showed that students demonstrated several CT abilities while working on their projects. The results suggest that a model like the one designed for this study, with a clear plan using a limited set of tools, can help and encourage teachers to integrate programming and CT in a cross-curricular manner [41]. Doing so, teachers move towards becoming "maker teachers" [37] while letting students learn basic programming skills and approach different aspects of CT while planning and designing their projects. Nevertheless, the model does not necessarily teach programming per se as, depending on the artefact designed and created, the programming needed can range from only a few blocks to long programs.

Our work contributes to current research on CT and STEAM in K-12 education, both by presenting the cross-curricular model and by describing how students and teachers experience and approach such a project. In addition, the results of our study give insight into which aspects of CT students experience less often, and that may therefore need to be made more explicit when planning and implementing such projects. A similar approach can be used when evaluating future cross-curricular implementations of CT, programming, STEAM and making in the classroom.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Funding

This research presented in this article was made possible with funding from the Technology Industries of Finland Centennial Foundation.

Acknowledgments

The authors want to thank Jenni Airola for helping out with the data collection and the analysis during the project.

References

[1] Arfé B, Vardanega T, Ronconi L. The effects of coding on children's planning and inhibition skills. Comput Educ 2020;148. https://doi.org/10.1016/j.compedu.2020.103807.

[2] Bakala E, Gerosa A, Hourcade JP, Tejera G. Preschool children, robots, and computational thinking: a systematic review. Int J Child Comput Interact 2021; 29. https://doi.org/10.1016/j.ijcci.2021.100337.

[3] Barefoot Computing. *The Computational Thinker: concepts and Approaches,* 2014. http://barefootcas.org.uk. Accessed on June 21, 2021.

[4] Barr V, Stephenson C. Bringing computational thinking to K-12: what is involved and what is the role of the computer science education community. ACM Inroads 2011;2(1):48–54. https://doi.org/10.1145/1929887.1929905.
[5] Bell T, Vahrenhold J. CS unplugged—how is it used, and does it work? In: Böckenhauer HJ, Komm D, Unger W, editors. Adventures between lower bounds and higher altitudes. *Lecture notes in computer science*. Cham: Springer; 2018. https://doi.org/10.1007/978-3-319-98355-4_29. vol 11011.
[6] Bienkowski M, Snow E, Rutstein D, Grover S. *Assessment design patterns for computational thinking practices in secondary computer science: a first look* (SRI technical report). Menlo Park, CA: SRI International; 2015. https://pact.sri.com/downloads/Assessment-Design-Patterns-for-Computational%20Thinking-Practices-Secondary-Computer-Science.pdf. Accessed on June 21, 2021.
[7] Bocconi, S., Chioccariello, A., Kampylis, P., Dagienė, V., Wastiau, P., Engelhardt, K., Earp, J., Horvath, M.A., Jasutė, E., Malagoli, C., Masiulionytė-Dagienė, V. and Stupurienė, G., Reviewing computational thinking in compulsory education, Inamorato Dos Santos, A., Cachia, R., Giannoutsou, N. and Punie, Y. editor(s), Publications Office of the European Union, Luxembourg, JRC128347, 2022. https://doi.org/10.2760/126955.
[8] Borg R, Porko-Hudd M, Hartvik J. Introducing maker-inspired technology in the Finnish craft subject –a case study from one school. Techne Series A 2021;28(2): 313–21. https://doi.org/10.7577/TechneA.4418.
[9] Brennan K, Resnick M. New frameworks for studying and assessing the development of computational thinking. In: Proceedings of the 2012 Annual Meeting of the American Educational Research Association; 2012.
[10] Caeli EN, Yadav A. Unplugged approaches to computational thinking: a historical perspective. TechTrends 2020;64(1):29–36. https://doi.org/10.1007/s11528-019-00410-5.
[11] Clarke D, Hollingsworth H. Elaborating a model of teacher professional growth. Teach Teacher Educ 2002;18(8):947–67. https://doi.org/10.1016/S0742-051X(02)00053-7.
[12] Chen G, Shen J, Barth-Cohen L, Jiang S, Huang X, Eltoukhy M. Assessing elementary students' computational thinking in everyday reasoning and robotics programming. Comput Educ 2017;109:162–75. https://doi.org/10.1016/j.compedu.2017.03.001.
[13] Chevalier M, Giang C, Piatti A, Mondada F. Fostering computational thinking through educational robotics: a model for creative computational problem solving. Int J STEM Educ 2020;7(39). https://doi.org/10.1186/s40594-020-00238-z.
[14] Chi MT. Quantifying qualitative analyses of verbal data: a practical guide. J Learn Sci 1997;6(3):271–315. https://doi.org/10.1207/s15327809jls0603_1.
[15] Cohen L, Manion L, Morrison K. Research methods in education. 6thEdition. London: Routledge; 2007.
[16] CSTA & ISTE. *Computational Thinking, Leadership Toolkit*. https://cdn.iste.org/www-root/2020-10/ISTE_CT_Leadership_Toolkit_booklet.pdf. 2011. Accessed on March 4, 2022.
[17] Del Olmo-Muñoz J, Cózar-Gutiérrez R, González-Calero JA. Computational thinking through unplugged activities in early years of primary education. Comput Educ 2020;150. https://doi.org/10.1016/j.compedu.2020.103832.
[18] Denzin NK. The research act: a theoretical introduction to sociological methods. New York: Praeger; 1978.
[19] de Paula BH, Burn A, Noss R, Valente JA. Playing Beowulf: bridging computational thinking, arts and literature through game-making. Int J Child Comput Interact 2018;16:39–46. https://doi.org/10.1016/j.ijcci.2017.11.003.
[20] dos Santos AI, Cachia R, Giannoutsou N, Punie Y. Reviewing Computational Thinking in Compulsory Education. State of play and practices from computing education. Joint Res Centre. Eur Commiss 2022:JRC128347. https://doi.org/10.2760/126955.
[21] European Schoolnet. *Computing our Future: computer Programming and Coding. Priorities, school curricula and initiatives across Europe.* Report. http://www.eun.org/documents/411753/817341/Computing+our+future_final_2015.pdf/d3780a64-1081-4488-8549-6033200e3c03. 2015. Accessed on June 21, 2021.
[22] Dong Y, Catete V, Jocius R, Lytle N, Barnes T, Albert J, Joshi D, Robinson R, Andrews A. PRADA: a practical model for integrating computational thinking in K-12 education. In: Proceedings of the 50th ACM Technical Symposium on Computer Science Education (SIGCSE '19). Association for Computing Machinery; 2019. p. 906–12. https://doi.org/10.1145/3287324.3287431.
[23] Drake SM, Reid JL. 21st century competencies in light of the history of integrated curriculum. Front Educ 2020. https://doi.org/10.3389/feduc.2020.00122. July.
[24] Fagerlund J, Häkkinen P, Vesisenaho M, Viiri J. Assessing 4th grade students' computational thinking through scratch programming projects. Inf Educ 2020;19(4):611–40. https://doi.org/10.15388/INFEDU.2020.27.
[25] Fagerlund J, Häkkinen P, Vesisenaho M, Viiri J. Computational thinking in programming with Scratch in primary schools: a systematic review. Comput Appl Eng Educ 2021;29(1):12–28. https://doi.org/10.1002/cae.22255.
[26] Fessakis G, Komis V, Mavroudi E, Prantsoudi S. Exploring the scope and the conceptualization of computational thinking at the K-12 classroom level curriculum. In: Khine M, editor. Computational thinking in the stem disciplines. Cham: Springer; 2018. https://doi.org/10.1007/978-3-319-93566-9_10.
[27] Fidai A, Capraro MM, Capraro RM. "Scratch"-ing computational thinking with Arduino: a meta-analysis. Think Skills Creativity 2020;38.
[28] Finnish National Agency of Education. *New national core curriculum for basic education: focus on school culture and integrative approach*. https://www.oph.fi/sites/default/files/documents/new-national-core-curriculum-for-basic-education.pdf. 2016. Accessed on June 29, 2021.
[29] Floyd LA. Computational Thinking in Teacher Education. In: Proceedings of the 51st ACM Technical Symposium on Computer Science Education (SIGCSE '20). Association for Computing Machinery; 2020. p. 1412. https://doi.org/10.1145/3328778.3372571.
[30] Gander W, Petit A, Berry G, Demo B, Vahrenhold J, McGettrick A, Boyle R, Drechsler M, Mendelson A, Stephenson C, Ghezzi C, Meyer B. Informatics education: Europe cannot afford to miss the boat. In: Report of the joint Informatics Europe & ACM Europe Working Group on Informatics Education; 2013.
[31] Grover S. Assessing algorithmic and computational thinking in K-12: lessons from a middle school classroom. In: Rich P, Hodges C, editors. Emerging research, practice, and policy on computational thinking. educational communications and technology: issues and innovations. Springer; 2017. p. 269–88. https://doi.org/10.1007/978-3-319-52691-1_17.
[32] Grover S, Pea R. Computational thinking in K–12: a review of the state of the field. Educ Researcher 2013;42(1):38–43. https://doi.org/10.3102/0013189X12463051.
[33] Hammond TC, Oltman J, Manfra MM. Geo-computational thinking in the third grade: making computational thinking trully "for everyone, everywhere". In: Proceedings of the 1st ACM SIGSPATIAL International Workshop on Geo-computational Thinking in Education (GeoEd'19)2. Association for Computing Machinery; 2019. p. 1–5. https://doi.org/10.1145/3356393.3365369.
[34] Harris A, de Bruin LR. Secondary school creativity, teacher practice and STEAM education: an international study. J Educ Change 2018;19:153–79. https://doi.org/10.1007/s10833-017-9311-2.
[35] Hill C. Computational thinking curriculum development for upper elementary school classes. In: Proceedings of the 10thAnnual Conference on International Computing Education Research; 2014. p. 151–2. https://doi.org/10.1145/2632320.2632327.
[36] Honey M, Kanter DE, editors. Design, make, play: growing the next generation of stem innovators. Abingdon: Routledge; 2013. https://doi.org/10.4324/9780203108352.
[37] Hughes J, Robb JA, Schira Hagerman M, Laffier J, Cotnam-Kappel M. What makes a maker teacher? Examining key characteristics of two maker educators. Int J Educ Res Open 2022;3. https://doi.org/10.1016/j.ijedro.2021.100118.
[38] Hus V, Grmek MI. Didactic strategies in early science teaching. Educ Stud 2011; 37(2):159–69. https://doi.org/10.1080/03055698.2010.506336.
[39] Hsu Y, Baldwin SJ, Ching Y-H. Learning through making and maker education. TechTrends 2017;61:589–94. https://doi.org/10.1007/s11528-017-0172-6.
[40] Hsu T-C, Chang S-C, Hung Y-T. How to learn and how to teach computational thinking: suggestions based on a review of the literature. Comput Educ 2018;126: 296–310. https://doi.org/10.1016/j.compedu.2018.07.004.
[41] Hsu YC, Ching YH, Baldwin S. Physical computing for STEAM education: maker-educators' experiences in an online graduate course. J Comput Math Sci Teach 2018;37(1):53–67. Waynesville, NC USA: Association for the Advancement of Computing in Education (AACE), https://www.learntechlib.org/primary/p/181338/. Waynesville, NC USA: Association for the Advancement of Computing in Education (AACE).
[42] Israel-Fishelson R, Hershkovitz A, Eguíluz A, Garaizar P, Guenaga M. The associations between computational thinking and creativity: the role of personal characteristics. J Educ Comput Res 2021;58(8):1415–47. https://doi.org/10.1177/0735633120940954.
[43] ISTE. ISTE Standards for Students. https://www.iste.org/standards/for-students. 2021. Accessed on June 26, 2021.
[44] Jenkins C. Poem generator: a comparative quantitative evaluation of a microworlds-based learning approach for teaching English. Int J Educ Dev Using Inf Commun Technol 2015:153–67.
[45] Jones C. Interdisciplinary approach – advantages, disadvantages, and the future benefits of interdisciplinary studies. ESSAI 2009;7. Article 26, http://dc.cod.edu/essai/vol7/iss1/26. Accessed on March 3, 2022.
[46] Jones WM, Cohen JD, Schad M, Caratachea M, Smith S. Maker-centered teacher professional development: examining K-12 teachers' learning experiences in a commercial makerspace. TechTrends 2020;64(1):37–49. https://doi.org/10.1007/s11528-019-00425-y.
[47] Juškevičienė A, Stupurienė G, Jevsikova T. Computational thinking development through physical computing activities in STEAM education. Comput Appl Eng Educ 2020;29(1):175–90. https://doi.org/10.1002/cae.22365.
[48] Kafai YB, Burke Q. The social turn in K-12 programming: moving from computational thinking to computational participation. In: *Proceeding of the 44th ACM technical symposium on Computer science education* (SIGCSE '13). New York, NY, USA: Association for Computing Machinery; 2013. p. 603–8. https://doi.org/10.1145/2445196.2445373.
[49] Kafai YB, Fields DA, Searle KA. Electronic textiles as disruptive designs: supporting and challenging maker activities in schools. Harv Educ Rev 2014;84(4):532–56.
[50] Kurti RS, Kurti DL, Fleming L. The philosophy of educational makerspaces. Teacher Librarian 2014;41(5):8–11.
[51] Kong S-C, Lai M, Sun D. Teacher development in computational thinking: design and learning outcomes of programming concepts, practices and pedagogy. Comput Educ 2020;151:1–19. https://doi.org/10.1016/j.compedu.2020.103872.
[52] Krapp A, Prenzel M. Research on interest in science: theories, methods, and findings. Int J Sci Educ 2011;33(1):27–50. https://doi.org/10.1080/09500693.2010.518645.
[53] Kukul V, Karataş S. Computational thinking self-efficacy scale: development, validity and reliability. Inf Educ 2019;18:151–64. https://doi.org/10.15388/infedu.2019.07.
[54] Lamprou A, Repenning A. Teaching how to teach computational thinking. In: Proceedings of the 23rd Annual ACM Conference on Innovation and Technology

in Computer Science Education (ITiCSE 2018). New York, NY, USA: Association for Computing Machinery; 2018. p. 69–74. https://doi.org/10.1145/3197091.3197120.
[55] Lee I, Grover S, Martin F, Pillai S, Maly-Smith J. Computational thinking from a disciplinary perspective: integrating computational thinking in K-12 science, technology, engineering, and mathematics education. J Sci Educ Technol 2020; 29:1–8. https://doi.org/10.1007/s10956-019-09803-w.
[56] Lindberg L, Fields DA, Kawai YB. Maker education: conceal/reveal of personal, artistic and computational dimensions in high school student projects. Front Educ 2020;5. https://doi.org/10.3389/feduc.2020.00051.
[57] Lye SY, Koh JHL. Review on teaching and learning of computational thinking through programming: what is next for K-12? Comput Human Behav 2014;41: 51–61. https://doi.org/10.1016/j.chb.2014.09.012.
[58] Malyn-Smith J, Lee I. Application of the occupational analysis of computational thinking-enabled STEM professionals as a program assessment tool. J Comput Sci Educ 2012;3(1):2–10. https://doi.org/10.22369/ISSN.2153-4136/3/1/1.
[59] Martin L. The promise of the maker movement for education. J Pre-College Eng Educ Res (J-PEER) 2015;5(1):30–9. https://doi.org/10.7771/2157-9288.1099.
[60] Martin F. Rethinking computational thinking. The advocate. ACM; 2018. February.
[61] Martínez Moreno J, Santos P, Hernandez-Leo D. Maker education in primary education: changes in students' maker-mindset and gender differences. In: Ninth International Conference on Technological Ecosystems for Enhancing Multiculturality (TEEM'21). New York, NY, USA: ACM; 2021. p. 120–5.
[62] Metcalf SJ, Reilly JM, Jeon S, Wang A, Pyers A, Brennan K, Dede C. Assessing computational thinking through the lenses of functionality and computational fluency. Comput Sci Educ 2021;31(2):199–223. https://doi.org/10.1080/08993408.2020.1866932.
[63] Michaeli T, Romeike R. Improving debugging skills in the classroom: the effects of teaching a systematic debugging process. In: Proceedings of the 14th Workshop in Primary and Secondary Computing Education (WiPSCE'19). 15. New York, NY, USA, Article: Association for Computing Machinery; 2019. p. 1–7. https://doi.org/10.1145/3361721.3361724.
[64] Moreno-León J, Robles G. Code to learn with Scratch? A systematic literature review. In: Proceedings of IEEE Global Engineering Education Conference; 2016. p. 150–6. https://10.1109/EDUCON.2016.7474546.
[65] Nouri J, Zhang L, Mannila L, Norén E. Development of computational thinking, digital competence and 21st century skills when learning programming in K-9. Educ Inquiry 2020;11(1):1–17. https://doi.org/10.1080/20004508.2019.1627844.
[66] OECD. The Future of Education and Skills. *Education 2030*. https://www.oecd.org/education/2030/E2030%20Position%20Paper%20(05.04.2018).pdf. 2019. Accessed on June 15, 2021.
[67] Oliver KM. Professional development considerations for makerspace leaders, part one: addressing "what?" and "why?". TechTrends 2016;60:160–6. https://doi.org/10.1007/s11528-016-0028-5.
[68] Papert S. Mindstorms: children, computers and powerful ideas. New York, NY: Basic Books; 1980.
[69] Parsazadeh N, Cheng P-Y, Wu T-T, Huang Y-M. Integrating computational thinking concept into digital storytelling to improve learners' motivation and performance. J Educ Comput Res 2021;59(3):470–95. https://doi.org/10.1177/0735633120967315.
[70] Pekrun R, Goetz T, Titz W, Perry RP. Academic emotions in students' self-regulated learning and achievement: a program of qualitative and quantitative research. Educ Psychol 2002;37(2):91–105. https://doi.org/10.1207/S15326985EP3702_4.
[71] Pekrun R, Linnenbrink-Garcia L, editors. International handbook of emotions in education. 1st edition. London: Routledge; 2014.
[72] Peppler K, Davis H. Arts and learning: a review of the impact of arts and aesthetics on learning and opportunities for further research. In: Proceedings of the 2010 International Conference of the Learning Sciences (ICLS); 2010.
[73] Perignat E, Katz-Buonincontro J. STEAM in practice and research: an integrative literature review. Thinking Skills Creativity 2019;31:31–43. https://doi.org/10.1016/j.tsc.2018.10.002.
[74] Petrie C. Interdisciplinary computational thinking with music and programming: a case study on algorithmic music composition with Sonic Pi. Comput Sci Educ 2021. https://doi.org/10.1080/08993408.2021.1935603.
[75] Piro J. Going from STEM to STEAM. Educ Week 2010;29(24):28–9.
[76] Rich PJ, Mason SL, O'Leary J. Measuring the effect of continuous professional development on elementary teachers' self-efficacy to teach coding and computational thinking. Comput Educ 2021;168. https://doi.org/10.1016/j.compedu.2021.104196.
[77] Rich KM, Yadav A, Larimore RA. Teacher implementation profiles for integrating computational thinking into elementary mathematics and science instruction. Educ Inf Technol 2020;25:3161–88. https://doi.org/10.1007/s10639-020-10115-5.
[78] Román-González M, Pérez-González J-C, Jiménez-Fernández C. Which cognitive abilities underlie computational thinking? Criterion validity of the computational thinking test. Comput Hum Behav 2017;72:678–91. https://doi.org/10.1016/j.chb.2016.08.047.
[79] Royal Society. *After the reboot: computing education in UK schools*. Report. https://royalsociety.org/~/media/events/2018/11/computing-education-1-year-on/after-the-reboot-report.pdf. 2017. Accessed on March 2, 2022.
[80] Ruthmann A, Heines JM, Greher GR, Laidler P, Saulters C. Teaching computational thinking through musical live coding in scratch. In: Proceedings of the 41st ACM technical symposium on Computer science education (SIGCSE '10). Association for Computing Machinery; 2010. p. 351–5. https://doi.org/10.1145/1734263.1734384.
[81] Sands P, Yadav A, Good J. Computational thinking in K-12: in-service teacher perceptions of computational thinking. In: Khine M, editor. Computational thinking in the stem disciplines. Cham: Springer; 2018. https://doi.org/10.1007/978-3-319-93566-9_8.
[82] Schleicher A. World class: how to build a 21st-Century school system. OECD; 2018. https://www.oecd.org/education/world-class-9789264300002-en.htm. Accessed on March 28, 2022.
[83] Shute VJ, Sun C, Asbell-Clarke J. Demystifying computational thinking. Educ Res Rev 2017;22:142–58. https://doi.org/10.1016/j.edurev.2017.09.003.
[84] Sun L, Hu L, Zhou D. Which way of design programming activities is more effective to promote K-12 students' computational thinking skills? A meta-analysis. J Comput Assist Learn 2021;37(4):1048–62. https://doi.org/10.1111/jcal.12545.
[85] Tang X, Yin Y, Lin Q, Hadad R, Zhai X. Assessing computational thinking: a systematic review of empirical studies. Comput Educ 2020;148:1–22. https://doi.org/10.1016/j.compedu.2019.103798.
[86] Tanhua-Piiroinen E, Kaarakainen S-S, Kaarakainen M-T, Viteli J. *Digiajan peruskoulu II* (Primary school in the digital age II). In: Final report, Finnish Ministry of Education and Culture; 2020. http://urn.fi/URN. ISBN:978-952-263-823-6. Accessed on March 28, 2022.
[87] Theodoropoulos A. Participatory design and participatory debugging: listening to students to improve computational thinking by creating games. Int J Child Comput Interact 2022;34. https://doi.org/10.1016/j.ijcci.2022.100525.
[88] Tikva C, Tambouris E. Mapping computational thinking through programming in K-12 education: a conceptual model based on a systematic literature review. Comput Educ 2021;162. https://doi.org/10.1016/j.compedu.2020.104083.
[89] Tsai M-J, Liang J-C, Hsu C-Y. The computational thinking scale for computer literacy education. J Educ Comput Res 2021;59(4):579–602. https://doi.org/10.1177/0735633120972356.
[90] Vahrenhold J, Caspersen M, Berry G, Gal-Ezer J, Kölling M, McGettrick A, Nardelli E, Pereira C, Westermeier M. Informatics education in Europe: are we all in the same boat. ACM Inf Europe 2017. https://kclpure.kcl.ac.uk/portal/en/publications/informatics-education-in-europe-are-we-all-in-the-same-boat(f83a8e4d-6f96-4d4a-bc5a-b1c676ee6ff1).html. Accessed on March 20, 2022.
[91] Veerasamy AK, Larsson P, Apiola M-V, D'Souza D, Laakso M-J. Pedagogical approaches in computational thinking-integrated STEAM learning settings: a literature review. In: Arai K, editor. Intelligent computing. lecture notes in networks and systems. Cham: Springer; 2021. https://doi.org/10.1007/978-3-030-80129-8_27. vol 285.
[92] Vinnervik P. Implementing programming in school mathematics and technology: teachers' intrinsic and extrinsic challenges. Int J Technol Des Educ 2022;32: 213–42. https://doi.org/10.1007/s10798-020-09602-0.
[93] Vuorikari R, Kluzer S, Punie Y. DigComp 2.2: the digital competence framework for citizens – with new examples of knowledge, skills and attitudes. In: EUR 31006 EN, Publications Office of the European Union, Luxembourg; 2022. https://doi.org/10.2760/490274. 2022, JRC128415.
[94] Waterman KP, Goldsmith L, Pasquale M. Integrating computational thinking into elementary science curriculum: an examination of activities that support students' computational thinking in the service of disciplinary learning. J Sci Educ Technol 2020;29:53–64. https://doi.org/10.1007/s10956-019-09801-y.
[95] Weintrop D, Beheshti E, Horn M, Orton K, Jona K, Trouille L, Wilensky U. Defining computational thinking for mathematics and science classrooms. J Sci Educ Technol 2016;25(1):127–47. https://doi.org/10.1007/s10956-015-9581-5.
[96] Wing J. Computational thinking. Commun ACM 2006;49(3):33–5. https://doi.org/10.1145/1118178.1118215.
[97] Wing, J. (2010). *Computational thinking: what and why?* https://www.cs.cmu.edu/~CompThink/resources/TheLinkWing.pdf. Accessed on June 7, 2021.
[98] Wong T. Makerspaces take libraries by storm. Library Media Connect 2013;31(6): 34–5.
[99] Wong GKW, Jiang S. Computational thinking education for children: algorithmic thinking and debugging. In: 2018 IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE); 2018. p. 328–34. https://doi.org/10.1109/TALE.2018.8615232.
[100] Wu SPW, Peel A, Bain C, Anton G, Horn M, Wilensky U. Workshops and Co-design can help teachers integrate computational thinking into their K-12 STEM classes. In: Proceedings of International Conference on Computational Thinking Education 2020. Hong Kong: The Education University of Hong Kong; 2020. p. 63–8.
[101] Yadav A, Good J, Voogt J, Fisser P. Computational thinking as an emerging competence domain. editor. In: Mulder M, editor. Competence-based vocational and professional education. Springer; 2017. p. 1051–67. https://doi.org/10.1007/978-3-319-41713-4_49.
[102] Yadav A, Hong H, Stephenson C. Computational thinking for all: pedagogical approaches to embedding 21st century problem solving in K-12 classrooms. TechTrends 2016;60:565–8. https://doi.org/10.1007/s11528-016-0087-7.
[103] Zhang L, Nouri J. A systematic review of learning computational thinking through Scratch in K-9. Comput Educ 2019;141. https://doi.org/10.1016/j.compedu.2019.103607.
[104] Heintz F., Mannila L., Färnqvist T. A review of models for introducing computational thinking, computer science and computing in K-12 education, 2016 IEEE Frontiers in Education Conference (FIE), Erie, PA, USA, 2016, pp. 1-9, doi: 10.1109/FIE.2016.7757410.

[105] Garcia-Peñalvo FJ. What computational thinking is. J Inf Technol 2016;9(3): v–viii.
[106] Linda Mannila, Lars-Åke Nordén, and ArnoldPears. 2018. Digital Competence, Teacher Self-Efficacy and Training Needs. In Proceedings of the 2018 ACM Conference on International Computing Education Research (ICER '18). Association for Computing Machinery, New York, NY, USA, 78–85. doi:10.1145/3230977.3230993.
[107] Mannila L. Digital kompetens i Svenskfinland – nulägesanalys och goda modeller. Svenska Kulturfonden; 2021.
[108] Mannila L. Integrating Programming in Other Subjects at Primary Level: Tool, Glue or Ideation. In: Keane T, Fluck AE, editors. Teaching Coding in K-12 Schools. Cham: Springer; 2023. https://doi.org/10.1007/978-3-031-21970-2_9.
[109] Mannila L, Heintz F, Kjällander S, Åkerfeldt A. Programming in Primary School: Towards a Research-Based Assessment Instrument. In: WiPSCE; 2020. https://doi.org/10.1145/3421590.3421598.
[110] Mannila L, Dagiene V, Demo B, Grgurina N, Mirolo C, Rolandsson L, Settle A. Computational Thinking in K-9 Education. New York, NY, USA: ACM; 2014. p. 1–29. https://doi.org/10.1145/2713609.2713610.
[111] Kjällander S, Åkerfeldt A, Mannila L, Parnes P. Makerspaces across settings: Didactic design for programming in formal and informal teacher education in the nordic countries. J Digit Learn Teach Educ 2018;34:18–30. https://doi.org/10.1080/21532974.2017.1387831.
[112] Strickland C., Rich K.M., Eatinger D., Lash T., Isaacs A., Israel M., Franklin D. Action Fractions: The Design and Pilot of an Integrated Math+CS Elementary Curriculum Based on Learning Trajectories. In Proceedings of the 52nd ACM Technical Symposium on Computer Science Education (SIGCSE '21). Association for Computing Machinery, New York, NY, USA, 2021, 11491155. doi:10.1145/3408877.3432483.