Baltic J. Modern Computing, Vol. 12 (2024), No. 2, 189-221
https://doi.org/10.22364/bjmc.2024.12.2.05

Fostering Computational Thinking in Compulsory Education in Europe: A Multiple Case Study

Valentina DAGIENĖ[1], Panagiotis KAMPYLIS[2],
Nikoleta GIANNOUTSOU[3], Katja ENGELHARDT[4],
Chiara MALAGOLI[2], Stefania BOCCONI[2]

[1] Vilnius University Institute of Educational Sciences, Universiteto str. 9, 01513 Vilnius, Lithuania
[2] Italian National Research Council – Institute for Educational Technology, Via de Marini, 6, 16149 Genova, Italy
[3] European Commission, Joint Research Centre, Calle Inca Garcilaso, 3, 41092, Seville, Spain
[4] European Schoolnet, Rue de Trèves, 61 (3rd floor), 1040 Brussels, Belgium

valentina.dagiene@mif.vu.lt, panagiotis.kampylis@itd.cnr.it,
nikoleta.giannoutsou@ec.europa.eu, katja.engelhardt@eun.org,
chiara.malagoli@itd.cnr.it, stefania.bocconi@itd.cnr.it,

ORCID 0000-0002-3955-4751, ORCID 0000-0003-1760-0347,
ORCID 0000-0002-1085-3854, ORCID 0000-0002-9916-3015,
ORCID 0000-0001-6872-2315, ORCID 0000-0001-6352-7414

Abstract. Many European countries have revised their curricula in recent years, introducing basic computer science concepts. This has paved the way for developing students' computational thinking (CT) skills. Despite increasing uptake, several issues and challenges are emerging for the effective integration of CT skills in compulsory education. The purpose of this paper is to examine the rationale for integrating CT skills in the European primary and lower secondary education curricula and the enablers and barriers to teaching and assessing these skills. A wide range of evidence was collected through three multiple-case studies involving 40 semi-structured interviews with experts, policymakers, school leaders and teachers, and 10 focus groups with students. Thematic analysis was conducted on 3,424 excerpts through NVivo to identify recurring codes and themes regarding implementation of CT skills within the curriculum in nine European countries and to explore commonalities and patterns across the cases. A common enabler is adopting appropriate measures for creating guidelines, learning materials and a large-scale professional development program. On the barrier side, the lack of qualified teachers, lack of quality materials and the challenges related to large-scale upskilling are shared.

Keywords: computational thinking (CT), compulsory education, computer science education, computing education, informatics education, multiple-case study

1. Introduction

1.1. Description of the Research Problem

Over the last fifteen years, computational thinking (CT) has attracted much attention from researchers in various areas and education policymakers. As a result, researchers tend to draw from various proposed CT definitions when conducting their investigations. Similarly, education policies also reflect this diversity in perspectives (Hsu et al., 2019).

The latest systematic reviews (e.g., Tikva and Tambouris, 2021; Lee et al., 2022) group CT definitions into two broad categories: 1) domain-specific and 2) domain-general. The first category focuses on problem-solving in Computer Science (CS) and is essentially dedicated to programming. The second category covers general problem-solving in various everyday life activities, with a particular emphasis on integrating CT into the learning process. Román-González et al. (2017) added operational definitions as a third category. Operational definitions break down CT into fundamental concepts -like algorithmic thinking, abstraction, decomposition, automation, and generalization- and related competences -like the creation of artefacts, collaboration, creativity, debugging, problem-solving of open-ended tasks, etc. (Grover and Pea, 2018). CT is considered a key competence for all citizens enabling them to deal with societal challenges. Furthermore, CT is interweaved with creativity in solving problems and applying innovative approaches in STEM and other disciplines, becoming a crucial element of education.

Programming is essential for a deeper understanding and development of CT. CT, on the other hand, integrates programming in a broader approach to solving problems (Metcalf et al., 2021; Tikva and Tambouris, 2021). The borders distinguishing the two are blurred: Programming is not necessarily part of CT. However, when programming is included in CT activities, then it shapes the learning process in a distinct way in the sense that it evolves through a continuous interaction between a) the formulation of a problem solution that a computer program can execute and b) the interpretation of immediate formative feedback generated by the execution of the program (Webb et al., 2017). Still, CT is broader than programming because it involves computationally inspired problem-solving in disciplines beyond CS (Hazzan et al., 2020). As a result, in studies exploring the teaching and learning of CT with computer use, CT definitions mainly deal with programming skills (Taslibeyaz et al., 2020). On the other hand, thinking skills (e.g., problem-solving, systemic thinking) are given more prominence (Fessakis and Prantsoudi, 2019; Sáez-López et al., 2016; Upadhyaya et al., 2020) in studies that explore the development of CT skills without using computers.

The above analysis shows two main trends in the discussion about CT, which are broadly defined by the inclusion or not of programming in CT. Jocius et al. bypass this dilemma by proposing a holistic approach when discussing the role of CT in general education (2020, p. 6): "the value of computational thinking is not just as an isolated concept that relates to computer science, but also as a way to enhance and support more complex discipline-specific and interdisciplinary understandings". According to Kale et al. (2018, p. 575), teaching CT should "entail the knowledge of using computational thinking tools (technology), knowing which instructional strategies to use to teach computational thinking and the subject matter (pedagogy), and understanding computational thinking and the subject matter (content)." This conceptualization of CT has important implications for teaching because it is envisaged to include three interconnected elements: technology, pedagogy and content. Technology involves knowledge of CT tools. Pedagogy involves knowledge and application of instructional strategies which cut across CT and the subject matter. Finally, content encompasses knowledge of CT and the subject matter. Hsu et al. (2019) draw on the results from their wide-ranging international review of CT policy-making in education to conclude that the different frameworks for understanding CT reflect the variety of ways CT education policies have been envisioned globally.

CT traces its roots back to CS development through decades. In the 1950s, CS programs were introduced to universities to meet a rising demand for learning new technology. Later, when many people, especially policymakers, saw how computing intervened in everyday work and home life, CS education started to be introduced to schools. Nevertheless, this movement was not easy. Then, CT was proposed with a focus on understanding computing. As a result, the CT movement has turned the corner on problem-solving using concepts and strategies most closely related to CS. CT is conceptualized as the set of thinking skills computer scientists use to address a broad range of problems (Denning and Tedre, 2022; Grover and Pea, 2018).

However, implementing the policies into practice is a complex issue with many dimensions being shaped but also shaping the conceptualization of CT. Specifically, research exploring the integration of CT in primary and secondary education, including teacher training, can be grouped into the following categories (see Table 1):

- CT integration in different education levels (including early childhood education, e.g. Manches and Plowman, 2017; Piedade and Dorotea, 2023);
- Curriculum development (Dagiene et al., 2021; Kong, 2016);
- Teaching and learning of CT which includes studies on
  - a) tools (Litts et al., 2021; Perin et al., 2023; Repenning et al., 2016),
  - b) activities (Olmo-Muñoz et al., 2020),
  - c) skills (Polat et al., 2021),
  - d) learning processes (Lai and Wong, 2022), and
  - e) assessment (Román-González et al., 2019);
- Teacher professional development (Yadav et al., 2016).

Table 1. Main theoretical contribution to CT education in schools

Category	Main theoretical concepts and principles
Category 1: General CT integration	deepening analysis of the relationships between CS and CT (Hazzan et al., 2020)
	adopted learning strategies and course categories of CT education (Hsu et al., 2018)
	instructional design, the development of Analysis, Design, Development, Implementation and Evaluation model (ADDIE) (Dagli and Tokmak, 2022)
	cognitive development and executive functions in connection to CT education in schools (Arfe et al., 2020)
	grouping CT skills into three large stages: defining the problem, solving the problem, and analysing the solution (Palts and Pedaste, 2020)
	focus on computational creativity (Israel-Fishelson, 2021)

Category 2: Curriculum development	development of informatics curriculum: the concept-context idea and separating fundamental concepts from the more volatile contexts (i.e., application domains and situations) (Barendsen et al., 2016); presentation and validation of progression frameworks for CT in schools based on computational agent theory (Curzon et al., 2019) support the implementation of the curriculum by delivering necessary guidance through delivery of instructional materials and activities (Kert et al., 2019) formation of a learning trajectory for math that extends into the area of CT (Niemelä et al., 2017) developing a framework for learning abstraction (Statter and Armoni, 2020)
Category 3: Teaching and learning of CT: tools, activities, skills, learning processes, and assessment	constructionism approach to delivery CT & CS concepts in schools; presentation of pedagogic approach in designing student-cantered conceptually focused microworlds as well as constructivism with its twin emphases on knowledge as constructed and on engagement in learning material for CS and CT (Csizmadia et al., 2019; Grover et al., 2019) frameworks for the relation between Scratch and CT development and for assessing areas in CT through Scratch projects (Fagerlund, et al., 2020; Wet at al., 2021) combining assessment tools for a comprehensive evaluation of CT education (Román-González et al., 2019; Yağcı, 2019) proposing a fundamental conceptual framework to analyse CT definitions and assessments, and providing a convenient assessment tool for future CT research and practices (Tsai et al., 2021) rationale of collaborative learning and especially of partial pair programming (Kong et al., 2018) assessment of modelling and simulation as intended learning outcomes for CT (Grgurina et al., 2018) conceptualizing programming empowerment as a part of CT (Zhang and Nouri, 2019) contribution to affordances of different rubric-based CT assessments of student programs (Metcalf et al., 2021) unplugged pedagogy supporting CT education and confirming that CT skills can be gained by using the unplugged coding activities (Huang and Looi, 2021; Tonbuloğlu and Tonbuloğlu, 2019)
Category 4: Teacher training	balance between the focus on CT concepts, teaching practices and identifying how CT can be embedded in other subjects (Paniagua and Istance, 2018) for successful teacher professional development, four factors are found: sustained periods of learning, active participation in training, connection to the school context, focus on pedagogical content knowledge (Kong et al., 2020) a new teacher training model: Code (Bootcamp), Connect (connecting disciplinary content and pedagogy to CT) and Create (the development of CT-infused learning segments) (Jocius et al., 2020) distinct approaches to integration CT in teacher training and related teaching methods (Sherwood et al., 2021)

While these studies provide valuable insights, there is little work bringing together the different dimensions of CT integration in compulsory education: 1) education policies around CT implemented in different education systems; 2) curricula integrating these educational policies; 3) insights from the implementation of CT in classes.

This study attempts a holistic understanding of CT integration in compulsory education by exploring the links between curricula integrating educational policies in different educational systems with teaching and assessment practices implemented in schools. To this end, we employed a multiple-case study methodology in compulsory education settings in nine European countries.

1.2. Study Goals

This paper is built on the (Bocconi et al., 2022) study. The paper aims to provide an up-to-date and wide-ranging overview of CT skills integrated into primary and lower secondary education in Europe by addressing two main research questions:

1. How are CT skills integrated into European primary and lower secondary education curricula?
2. What are the enablers and barriers to teaching and assessing CT skills in the European primary and lower secondary education curricula?

These questions are timely as many European countries have recently revised their compulsory education curricula introducing basic CS concepts for developing students' CT skills (Bocconi et al., 2022). The main assumption is that CT skills are part of young people's digital competences essential for living and working in our digital world. In other words, CT is considered a fundamental skill (or set of skills) not only for computer scientists but also for all citizens, and its development must start from an early age (European Commission, 2020). However, introducing CT skills into school curricula is a complex process requiring more evidence on what to be taught, when, and how.

2. Method

2.1. Multiple-Case Study Research

We adopted a multiple-case study (MCS) methodology (Yin, 2018) as it is an established research design for researching a particular phenomenon in-depth (Zach, 2006). MCS enables comparisons to determine whether emergent findings are distinctive for a single case or reliably replicated across multiple cases (Eisenhardt and Graebner, 2007). Also, the MCS methodology enables a deeper investigation of research questions, and it can provide reliable and robust evidence that can be used as a basis for theoretical elaborations (Baxter and Jack, 2008).

All the selected case studies fall within primary and lower secondary education (corresponding to ISCED 1 and ISCED 2 according to UNESCO, 2012) that in most European countries comprises compulsory education (European Commission et al., 2020) aiming to enable all students to develop their key competences.

To make sound cross-case comparisons, careful case selection was critical to guarantee that consistent and inconsistent results could emerge from the broad set of cases, allowing to make solid cross-case comparisons. For example, the selection of the nine cases is considered to be within the threshold recommended in Yin's replication strategy, i.e. 6-10 cases (Yin, 2018). The criteria for the selection were the following:

- CT adoption forming part of a curriculum development policy;

- maturity level, concerning the implementation of the initiative with a minimum of two years;
- diverse geographical coverage;
- availability of relevant documentation;
- education level coverage within compulsory education (i.e., primary and lower secondary);
- CT integration approach (i.e., CT as a distinct subject; as part of other subjects; as a cross-curriculum theme).

According to Bocconi et al. (2016), three different approaches typify CT skills integration in compulsory education curricula: 1) as a cross-curriculum theme; 2) embedded as an integral part of a distinct CS subject; 3) addressed within other subjects. These approaches for CT skills integration are confirmed by a recent survey of policy initiatives (Bocconi et al., 2022) and the literature (e.g. Eurydice, 2019; Syslo, 2020).

The choice between these different strategies at the national (and/or regional) level is determined to a large extent by the political mandate for the integration of CT skills in the specific country/region and by the organizational constraints governing how curriculum changes are implemented and rolled out within the education system in question. The choice of where and how to integrate CT into the curriculum is also driven by different motivations for why CT is being integrated. For example, the integration of CT as a cross-curriculum theme is usually underpinned by the rationale that CT can foster transversal skills relevant to and practically applicable to all subject areas. On the other hand, integrating CT as a distinct single subject is closely related to the introduction of CS Education. Finally, integrating CT as part of an already-existing subject is more likely due to policy constraints that could derive from the challenges of introducing a new subject area in an already crowded curriculum. Choosing which of the three approaches to adopt is also closely linked to the different education levels. In primary education (ISCED 1), subject-area distinctions are less marked and do not play a central role; teachers cover several subjects without necessarily having specialized expertise in each one. Hence, the cross-curriculum theme is a strategy more commonly adopted at this education level. Moving on to lower secondary (ISCED 2), distinct subjects start to play a central role and teachers are often recruited based on their qualifications in the specific subject area(s). At this education level, the policy choice to either embed CT as a separate subject or within existing subjects takes priority.

The three MCSs address how schools in nine European countries teach and assess CT skills, what tools teachers and learners use, and how teachers are upskilled. Each MCS comprises three distinct cases and represents one of the three approaches for integrating CT skills in the compulsory education curricula mentioned above:

- MCS1: CT as a cross-curriculum theme in primary education;
- MCS2: CT integrated as a separate subject in lower secondary education;
- MCS3: CT within other subjects in lower secondary education.

Within each MCS, two cases were selected as a literal replication of each other, i.e. leading to comparable results. In contrast, the third case was chosen as a theoretical replication, i.e., as a “sounding board” that confirms or contradicts the previous results (see Table 2).

Table 2. Yin's replication strategy adopted for the three multiple case study

Replication strategy	MCS1: CT as a cross-curriculum theme in primary education	MCS2: CT as a separate subject in lower-secondary education	MCS3: CT within other subjects in lower-secondary education
Literal replication	C1 – Lithuania	C4 – Croatia	C7 – France
	C2 – Norway	C5 – Poland	C8 – Finland
Theoretical replication	C3 – Slovakia	C6 – England (UK)	C9 – Sweden

MCS1 considers the implementation of CT as a cross-curriculum theme in primary education. We selected Lithuania and Norway as two cases that addressed CT as a cross-curriculum theme and as part of digital competence. We set those two cases against the case of Slovakia, which integrates CT into the curriculum as a distinct subject, namely Informatics.

In MCS2, we consider the implementation of CT as a distinct CS subject at the lower secondary level. We selected Poland and Croatia, which have a long-standing tradition of Informatics education, mostly at the upper secondary level and more recently extended to lower secondary and primary education. England (UK), as of September 2014, has implemented the Computing curriculum in primary and secondary schools.

MCS3 considers CT implementation in lower secondary schools in Finland, France and Sweden. In these three countries, CT is covered in Maths & Technology. However, Sweden also includes CT in Social Studies.

The phases of the adopted MCS research methodology are shown in Figure 1.

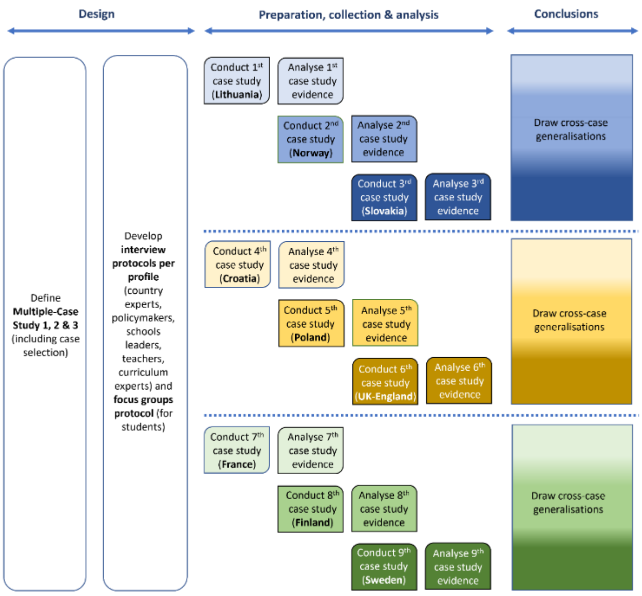

Figure 1. The design, preparation, data collection, analysis, and conclusions of the MCSs conducted

2.2. Data Collection

2.2.1. Identification Procedures

The data collection process elicited insights from key actors and perceived changes in their practice following their direct engagement in CT curriculum enactment. We conducted semi-structured interviews and focus groups examining in-depth the stakeholders' perspectives through open-ended questions (questions are presented in Appendix 1). Different but overlapping interview protocols were developed for policymakers, school leaders, teachers, MCS experts and for the students' focus groups covering a variety of topics such as CT definition, what are the enablers and barriers to teaching and assessing CT skills and what are the effective ways for teacher training.

Three protocols were developed for the semi-structured interviews with policymakers, teachers, and school leaders, and one for student focus groups. Following the approach proposed by Yin (2018), after the conclusion of each case, the interview protocols were reviewed and, when necessary, integrated with new questions deemed helpful in informing the following case interview round.

The interviews and focus groups were conducted by research team members using videoconferencing tools. Each interview lasted between 45 and 60 minutes, and all discussions were audio-recorded. Each recording was automatically transcribed verbatim using Nvivo12. Transcriptions in languages other than English were translated into English using an online automatic text translation system (deepl.com). The output was then validated for English-language congruency by a native English speaker. Finally, the respective interviewers checked all transcriptions to verify overall consistency with the original content.

Following the MCS methodology, data were analysed in two steps. The first step was to analyse all transcripts from an individual case to thoroughly understand how CT skills are integrated into the curriculum at hand. The emerging concepts progressively identified from the coded interviews were included in the analysis of the subsequent interviews to be coded. This systematic, incremental procedure was applied to analyse all interview transcripts. The second step was the cross-case analysis permitting the identification of similarities and differences between the three cases of each MCS.

2.2.2. Ethical Considerations

Gaining informed consent from each participant was a primary ethical concern, as was compliance with personal data management provisions, both being key to the General Data Protection Regulation (GDPR), which we followed. Therefore, informed consent was a precondition for conducting each interview, and involvement in the study was voluntary. Furthermore, participants were informed from the outset that they had the right to withdraw until the data analysis was completed. For students, consent forms were collected and retained by their respective schools, which were required to sign a declaration stating that both the students and parents had been duly informed and agreed to participate.

All personal information gained about participants and participant identities were fully protected and managed in compliance with GDPR provisions. Before and after each interview, interviewers assured participants of complete confidentiality.

Furthermore, in compliance with GDPR provisions, all the collected data was safely stored and will be duly deleted after five years.

2.3. Context and Participants

The criteria for selecting participants were: 1) the direct experience in implementing and enacting the CT curriculum; 2) the profile range (student, teacher, school leader, researcher, expert, policymaker).

Ninety-two subjects representing different stakeholder categories participated in the three MCSs. We conducted 40 semi-structured interviews with policymakers, experts, school leaders, and teachers and 10 focus groups with 52 students (see Table 3). Interviewees were also asked to provide documents and evidence of the CT skills integration in the curriculum, such as platforms and tools used, learning materials produced by teachers, or information on final exams or other assessments. In addition, a CT curriculum expert was interviewed for each multiple case (MCS1, MCS2 and MCS3).

Table 3. Interviews per profile, country and MCS

MCS	Country	Country experts	Policy makers	School leaders	Teachers	Focus Groups	Curriculum experts
MCS1	Lithuania	-	1	1	1	1	1
	Norway	-	1	1	1	1	
	Slovakia	-	1	1	1	1	
MCS2	Croatia	-	1	1	1	1	1
	Poland	1	1	1	1	1	
	UK-England	1	-	1	1	1	
MCS3	France	1	1	2	3	2	1
	Finland	1	1	1	2	1	
	Sweden	1	1	1	2	1	
Total		**5**	**8**	**10**	**14**	**10**	**3**

The participation of 92 subjects was reasonable, given the warning from Creswell and Gutterman (2019) that including many individuals can result in superficial perspectives rather than an in-depth picture. In addition, involving different stakeholder groups makes it possible to gather different viewpoints and insights from those involved in implementing and enacting the curriculum.

For each case study in MCS1 & MCS2, individual semi-structured interviews were conducted with one school leader, one teacher in charge of CT integration, and one policy maker. In addition, a cohort of 5-7 students participated in a focus group. In UK-England, we were not able to interview a policymaker. Instead, we interviewed a country expert involved in developing and writing the English Computing Curriculum.

Given that in MCS3 the CT integration approach within subjects involved both Mathematics & Technology in each of the three country cases, a teacher of each subject was interviewed. Furthermore, an expert involved in the definition and/or implementation of the CT curriculum at the national level was also interviewed in each case to better understand the role of CT in the two subjects.

The French case was scheduled before the end of the school year (beginning of July). As a risk management measure, we did contact two schools in May that agreed to participate only in the second half of June. Thus, in France, we took the opportunity to gain insights from two different school contexts.

2.4. Data Analysis

In our analysis, we employed two different methodologies. One was thematic analysis which supported us in examining in depth each case study, and the other was cluster analysis which allowed us to understand patterns, similarities and differences across the cases of each MCS.

2.4.1. Thematic Analysis

We followed Braun and Clarke's (2006) six steps for thematic analysis to extract and analyse precise information from the interviews and focus groups. We have adopted a mixed approach to coding, combining predefined codes related to the study's specific goals and research questions (structural codes) and codes that emerged from raw data processing (data-driven codes) (Ryan and Bernard, 2003). To begin the coding process, we defined an initial common set of provisional codes based on the study's research questions (structural coding schema). This schema comprised 48 codes grouped into twelve categories and was universally applied to codify the interview transcripts in each of the three multiple-case studies. Overall, 26 data-driven codes emerged while analysing the nine individual cases comprising the three multiple-case studies. These data-driven codes, which largely reflect the specificity of the multiple case studies from which they derived, expanded and enriched the set of codes under the 12 code categories identified in the initial structural code schema.

A systematic, step-by-step coding and analysis process was conducted on the entire set of interview transcripts in English using NVivo12, a widely adopted multilingual software package for processing and analysing data within qualitative research studies (Jackson and Bazeley, 2019). The coding process was performed cumulatively: the initial set of structural codes was progressively augmented with emergent data-driven codes, thus requiring the raw data to undergo rounds of re-processing. However, due to the time constrain of this study, only one iteration was performed on each document in the data set.

We used the complete coding scheme comprised of 48 predefined codes emerging from the literature and 26 from the data to identify themes regarding the integration and implementation of CT skills within the curriculum in nine European countries and examine commonalities and patterns across the cases.

The overall analysis of the 40 semi-structured interviews and the ten focus groups with students regarding implementing CT skills within the curriculum in nine European countries resulted in 3,424 excerpts coded through NVivo 12.

2.4.2. A Cluster Analysis

A cluster analysis was conducted to explore patterns within each MCS by identifying similarities and differences among the terms used, i.e., among insights highlighted by the various stakeholders. For each MCS, the cluster analysis was implemented in NVivo 12, clustering child codes by word similarity and including all child codes that registered at least two excerpts. Pearson correlation coefficient (r) was used as a similarity metric. The range of this coefficient is [-1, 1]; from least similar (r = -1) to most similar (r = 1). The data of each cluster analysis are graphically represented via dendrograms. As a built-out hierarchy of clusters, dendrograms show how the selected codes merge with others at certain distances. Interconnected codes are grouped together. The analysis results within each MCS show codes clustered by word similarity, with similar items clustered together on the same branch and different items positioned further apart (see Table 4).

To identify key findings from cluster analysis to compare with findings from coding the interviews, we examined sub-clusters where the Pearson correlation coefficient (r) among child codes was greater than 0.5. This led to selection of three sub-clusters for the MCSs where most of the correlation coefficient among child codes was strong (r > 0.7). The tree sub-clusters (see Figures 2, 3, and 4) comprise 19 structural codes; and four data-driven codes referring to 1,763 excerpts out of 3,424 coded in the thematic analysis (see Table 4). Results are presented in more detail in the next section.

Table 4. Codes and numbers of referring excerpts present in the selected sub-clusters

Codes	Description	MCS1 exc.	MCS2 exc.	MCS3 exc.	Total
Policymakers	Any reference to the involvement/contribution of policymakers at the local, regional, national or EU level	8	10	16	34
School leaders	Any reference to the involvement/contribution of the school leadership team (school leaders, school heads, school principals, head teachers and their deputies)	20	14	9	43
Teachers	Any reference to the involvement/contribution of teaching staff	90	9	13	112
CT concepts	Any reference to the various CT concepts	6	0	28	34
Progression	Any reference to curriculum progression	2	8	40	50
Pedagogies	Any input on the pedagogies adopted for teaching CT and related concepts	42	124	111	277
Learning Tools	Any input on the technologies and tools employed for teaching and learning CT and related concepts	75	46	64	185
Within-subjects	An instance when CT is part of one or more existing subjects such as Mathematic, Technology, etc.	17	5	34	56
Barrier	Any reference to challenges and inhibiting factors for the integration of CT in the curriculum	14	39	95	148

Enabler	Any reference to enablers and success factors for the integration of CT in the curriculum	14	27	42	83
Recommendation	Indications that may help to formulate policy recommendations	29	62	77	168
Curriculum	Any reference to CT in the curriculum (e.g., objectives, location)	19	80	64	163
Monitoring/Evaluation	Any reference to measures put in place for monitoring and evaluating the impact and sustainability of CT integration in the curriculum	5	5	13	23
Reform	Any reference to specific reforms enacted towards integration of CT in the curriculum	6	14	32	52
Gender	Any reference to ways of ensuring gender balance when implementing CT in the curriculum.	11	21	32	64
Fostering transversal skills	Any reference to transversal skills, such as creativity, that are fostered when implementing CT in the curriculum	3	1	8	12
Relationship with programming	Any reference to CT's relationship with programming	21	9	14	44
Recruitment	Any reference to teacher recruitment or retention strategies pertaining to CT	12	30	2	44
Training needs	Any reference to specific CT training needs that are not covered by existing training provisions	21	0	8	29
Teacher Support (emerged from MCS1)	Any teacher reference to how they supported CT teaching	15	NA	NA	15
Compulsory subject (emerged from MCS2)	Any references to the Computer Science/Informatics/Computing subject being compulsory (reasons, conditions)	NA	13	NA	13
Continuous Professional Development (CPD) (emerged from MCS2)	Any references to teachers' CPD need and provision	NA	68	NA	68
Numbers of hours per week dedicated to CT (emerged from MCS3)	Any reference to the estimated number of weekly hours dedicated to CT (e.g., by the teachers and/or the school leader)	NA	23	23	46
Total		**430**	**608**	**725**	**1,763**

3. Results

In this section, we report the key findings from the interviews and focus group analysis of each multiple case study summarized as enablers and barriers. In addition, the findings are compared with a cluster analysis of the transcripts implemented in NVivo 12.

3.1. Main Outcomes from MCS1: CT as a Cross-Curriculum Theme in Primary Education

A key takeaway from MCS1 is that CT, as a cross-curriculum theme or part of a separate CS/Informatics subject, is becoming an essential element of the primary school curriculum. Teaching CT-related topics at primary school is critical for students to understand the digital world in which they live and become fluent and creative with technology. The ministries of education, universities and other institutions actively supported the implementation of the Informatics curriculum at primary education through programs for teachers' professional development and the provision of educational resources. In Lithuania, following the General Teaching Plan for 2022 and 2023, all primary schools are encouraged to implement Informatics/CT in their classes. In primary education Informatics curriculum covers most of the major CT components. However, it is up to up to individual schools (and their teachers) to decide how and in which subjects to integrate the Informatics/CT components they choose to address.

The revised Norwegian national curriculum came into force in 2020. The country's Ministry of Education published a digitalization strategy for primary, secondary, and vocational education for 2017-2021, stating that CT and programming should be integrated into the curriculum. The new curriculum foresees the integration of CT and programming in Mathematics, Science, Arts & Crafts, and Music, both at primary and lower secondary levels. The curriculum emphasizes the development of algorithmic thinking skills by addressing the steps involved in solving problems through programming. Slovakia's comprehensive reform of its education system in 2008 brought significant changes to the content of all subjects in primary and secondary schools. One of the main changes was the introduction of Informatics as a compulsory separate subject in grades 3-4. The Case Study school teacher explained that pupils are introduced to basic computational concepts starting from simple computer instructions and then arranging these instructions into procedures and loops.

Primary school teachers in Lithuania, Slovakia and Norway mentioned learning approaches such as learning by doing, playful learning, learning from mistakes, and working in small groups. Specifically, teachers emphasized using hands-on, playful activities with programmable robots and visual block environments to introduce basic CS concepts to students. Students begin by giving physical and/or virtual objects sequences of instructions to perform when developing their projects. By controlling robots or constructing programs through these sequences of instructions, students gradually move from being passive technology consumers to active digital object creators. Primary school teachers also report how student-to-student and teacher-to-student discussions and interactions during the hands-on activities promote peer learning. For example, they encourage more advanced students to help classmates stuck somewhere. All three countries use a variety of educational tools for teaching CT/Informatics/programming. Some derive from abroad, while others have been specially designed and promoted by national institutions. However, learning by doing is the most commonly adopted methodology.

Primary teachers interviewed in the context of MCS1 pointed out that they mainly perform a formative assessment of CT skills. Assessment is based on teachers observing students while solving problems, developing projects, and learning by reflecting on mistakes. At the end of learning activities, group discussion encourages and allows students to express their thoughts and how they think and arrive at solutions. Teachers also employ quizzes (such as Bebras tasks), exercises and surveys to assess CT skills.

As seen in Table 5, one of the main challenges is the lack of qualified teachers encountered in two of the three CS. It is more likely for adequately trained and qualified teachers to be able to stimulate students to more profound CS learning. The literature also supports this finding. For example, the shortage of appropriately trained teachers across all education levels was mentioned most (83 times) in the CT Educational Policy Initiatives survey (Hsu et al., 2019) when respondents were asked to identify the main problems and challenges in integrating CT in schools.

Another barrier in the same two countries (i.e., Lithuania and Norway) relates to the lack of appropriate resources. It is interesting to see that the perceived enablers focus on how mitigating these problems, i.e. provision of teacher training by the ministries and/or universities. The findings from Slovakia are differentiated, but they demonstrate an important dimension of CT integration, which is related to the management of the material dimension of technology (taking care of the hardware, charging the devices) and is often overlooked. Finally, both Norway and Slovakia seem to place particular emphasis on the role of school leaders in promoting CT, pointing to the critical role of the school as an organization.

Table 5. Main enablers and barriers in implementing CT at the primary education level

Country	**Enablers**	**Barriers**
Lithuania	The policy introducing CT is well-accepted by teachers. The Ministry of Education and universities actively support the implementation of CT in schools by providing professional teacher training and developing educational resources.	Shortage of adequately trained teachers and educational resources. Lack of interesting activities for learning CT.
Slovakia	Many school leaders support the development of digital competence for students and teachers. Learners like programming physical and virtual robots working in groups.	In implementing the curriculum, many teachers focus on practical aspects at the expense of addressing CS basic principles and concepts. Lack of programming environments suitable for young children's cognitive abilities.
Norway	School leaders promote teacher participation in professional development. The Ministry of Education promotes engaging CT activities that foster learners' motivation through guidelines and teacher support.	Lack of qualified teachers at the primary level. Shortage of suitable materials for primary classes.

The cluster analysis by word similarities on the transcripts of the interviews and focus groups conducted in MCS1 revealed a significant correlation among the codes "Recruitment", "Teachers", "Barrier", and "Training needs". Observing the central part of the dendrogram (Figure 2), in fact, "Teachers," "Training needs", and "Barrier" are connected with a high correlation among sub-codes (r=.84). These sub-codes also emerged close in similarity with “Recruitment” (r=.75). This pattern reflects the interviewee’s understanding of the centrality of teachers’ recruitment and training regarding CT integration.

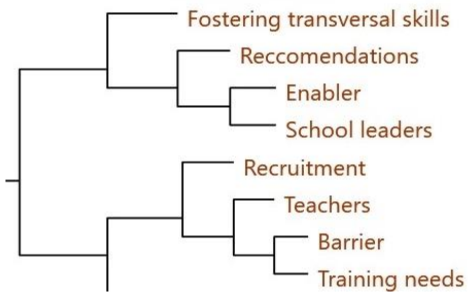

Figure 2. A selection of the cluster analysis hierarchy for CT as a cross-curriculum theme (MCS1)

A second pattern encompassed the codes “Fostering transversal skills”, “Enabler”, and “School leaders”, with “Enabler” and “School leaders” strongly correlating (r=.71). This pattern reflects interviewees’ understanding of CT as a critical dimension to ground and foster students’ transversal skills, including digital competence, where school leaders play a critical role in facilitating this synergy in implementing the curriculum.

3.2. Main Outcomes from MCS 2: CT Integrated as a Separate Subject in Lower-Secondary Education

In MCS2, the implementation of CT as a distinct CS subject at the lower secondary level was considered in Croatia, UK-England and Poland. In UK-England, the subject is referred to as ‘Computing’, Poland as ‘Computer Science’ and Croatia as ‘Informatics’. Unlike UK-England, Poland and Croatia have a long-lasting tradition of CS education. However, already in 2014, UK-England started implementing CT education in primary and secondary schools. The current CS curricula have been in place in Poland and Croatia since 2017.

Curricula in Croatia and Poland both focused on developing students’ problem-solving skills. The Polish CS curriculum aims to engage students to apply CT skills in problem-solving in all other school subjects. Computing education in UK-England aims to equip students ‘to use CT and creativity to understand and change the world’. Students should know the key concepts and principles of data, information, algorithms and computation, about functionality and processes in digital systems and how to connect this knowledge to programming. Students should also acquire better digital literacy.

All three CS curricula effectively combine elements of digital literacy and CS concepts. The Croatian curriculum, for instance, also aims for students to learn how to use technology in a practical, reasonable and respectful manner and consists of four domains: 1) CT and programming; 2) Information & Digital Technologies; 3) Digital Literacy & Communication; and 4) e-Society. CS, as a compulsory subject, is taught 1 or 2 hours per week; this time is insufficient for addressing all subject areas. More time is spent on teaching CS concepts at higher grades in Croatia and UK-England. Compulsory elements of CS are usually complemented with elective subjects and extra-curricular activities. There seems to be a strong rationale for teaching CT as part of a separate CS subject because of the underlying subject knowledge and foundational subject discipline best taught as part of a CS subject (e.g., concepts behind the internet, limitations of computers). All three curricula also highlight the importance of links with other subjects.

Regarding the implementation of curricula in classrooms, considerable differences exist between schools due to differences in digital infrastructure and the availability of qualified teachers. As CS curricula are generally open-ended and they focus on skills such as problem-solving, they invite the use of innovative pedagogical approaches that foster student autonomy. Examples include personalized learning and cooperative or collaborative learning. Such teaching requires teachers and students to be open-minded and flexible about the learning processes and their results. The Polish and Croatian curricula suggest that teachers start with visual languages in the younger grades and gradually move to textual languages. The key to successful CS education seems to enable students to solve real-world problems or create something new and find and correct mistakes in the process. Assessment should capture excellence without discouraging average students. Focus is devoted to formative assessment practices. The lack of qualified teachers emerged as the critical barrier to quality CS education (see Table 6). In UK-England and, to some extent, in Poland, curricula were introduced in schools before teachers were fully upskilled at scale. One hypothesis at the outset of MSC2 was that there might be differences in how CT and related activities are implemented in Poland and Croatia, which both have a long-standing tradition in CS, compared to England. One possible reason was that the subject CS already existed in the curriculum and finding a place in the curriculum for CS topics is often one of the main challenges. The second possible reason was that initial teacher training for CS was already in place. The cases of Poland and Croatia could rely on teachers already trained in CS. In the case of England, the fact that teachers with no CS background had to be prepared to teach the newly introduced Computing subject has been stressed as having constituted an enormous challenge. Nonetheless, in Poland and Croatia, introducing the current CS curricula remained a challenge since CS has been comprehensively integrated at all education levels, and its focus has been adjusted. The new curriculum necessitated large-scale CDP also for CS teachers. Finally, recruiting a sufficient number of CS teachers remains a challenge also in these countries.

Table 6. Main enablers and barriers in implementing CT as a separate subject at lower secondary school

Country	**Enablers**	**Barriers**
Croatia	Schools got internet connections and devices. Large-scale online training was offered before the curriculum started. Key features were learning communities, training on new topics every few weeks and continuous support. Teachers tried different teaching and assessment methods and created a shared base of learning resources.	The main problem is the lack of adequately trained teachers. This became most apparent when nformatics became an elective subject at all primary schools in 2020, and students have shown great interest. School heads need support to implement a whole school approach to teaching CT in Informatics and other subjects.

Poland	The renewal of the CS curriculum was part of a change in the core curriculum in 2017. The CS curriculum leaves freedom to teachers, for whom it is both an opportunity and a challenge. The curriculum includes guidelines, training materials and new computer science textbooks. Projects and extra-curricular activities remain essential. Piloting curricular changes with pilot schools first prepares a smooth curriculum roll-out. Robust monitoring and evaluation of the curriculum will be needed.	The key challenge is implementing teacher training on a large scale for all teachers. CS is a compulsory subject in all grades, but there are open questions about how schools implement the subject using tools and methods.
UK-England	Key enablers are quality teacher training, school hubs connecting different schools and specific measures to recruit new Computing teachers. Other enablers are sustainable funding and synergies between the Department of Education and other organizations (e.g., grassroots). The National Centre for Computing Education produced high-quality teaching resources and teacher training, with lesson plans, assessment exercises and teacher guidelines. Research on how CT skills are best taught and how CS is integrated into schools.	The main challenge is a lack of properly educated teachers. In the first phase of the roll-out of the Computing curriculum, not enough training opportunities were available to teachers. Computing education is compulsory until key stage 3. However, there are considerable differences in how schools implement the curriculum. School leaders and senior leadership teams often do not consider computing education a priority.

In conclusion, for high-quality teaching of CS, teachers need training and sufficient time to teach CT and related skills, but also more research and robust monitoring of the curriculum implementation are crucial. The cluster analysis applied to all interviews and focus groups conducted in MCS2 revealed a significant correlation among the codes “Teacher support” and “Continuous Professional Development (CPD)” (r=.84) connected to “Pedagogies” and “Recommendations” (r=.65) “.

In the branches below (Figure 3), “Number of hours per week dedicated to CT” has a significant correlation with “Curriculum” (r=.75), “Reform”, and “Learning tools” (r=.73). “Monitoring evaluation” and “Policy-makers” (r=.69) are connected in the upper branches to “School leaders” with a high coefficient of similarity (r=.68). Right above, “Relationship with programming” is connected to “Curriculum” (r=.67), and “Compulsory subject” is connected to “Curriculum” (r=.63) and “Number of hours per week dedicated to CT” (r=.53). This reflects the interviewees’ emphasis on the exchange of CT know-how, dedicated time and practices among teachers and school leaders to support and promote the integration of CT-related content in teaching and learning practices.

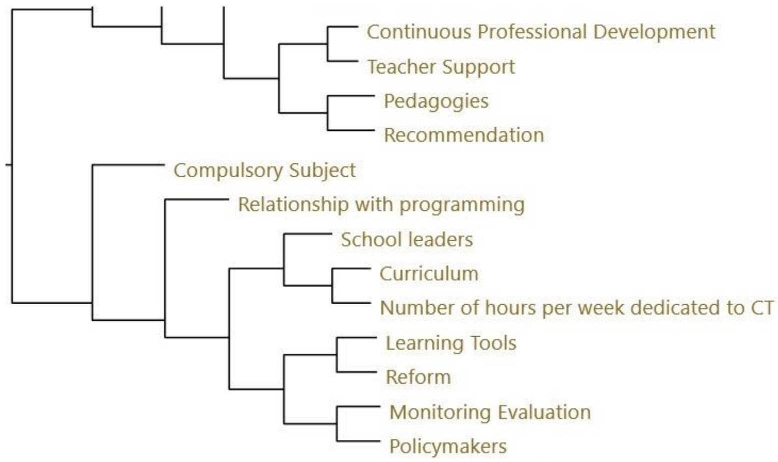

Figure 3. A selection of the cluster analysis hierarchy for CT as a separate subject (MCS2)

3.3. Main Outcomes from MCS 3: CT Within Other Subjects in Lower-Secondary Education

The MCS3 investigates how CT skills and related CS concepts have been integrated into lower secondary education within other subjects in Finland, France and Sweden. The first two countries have similar approaches focusing on introducing programming and algorithmic thinking, mainly in Mathematics and another subject: Craft in Finland and Technology in France (see Table 7). The Swedish curriculum introduces programming in Mathematics, Technology and Civics, focusing on the role and implications of programs and algorithms in digital technology for society and the individual. While we initially hypothesized that the inclusion of programming within Civics in Sweden could promote differences in the implementation compared to France and Finland, this is not the case. During the interviews, the Swedish policymaker and expert clarified that the major focus within Civics is not on programming practices but mostly on understanding the implications and roles of programs and digital technology for society and the individual. Therefore, students are not expected to build programs or use algorithms, although teachers can, in principle, also decide to propose such activities.

While the curricula in all three countries do not specify the amount of time for the added CS contents, the mathematics curriculum limits the time teachers can devote to algorithms and programming. The interviewed Mathematics teachers estimated that this time was less than 1 hour per week in all three countries. The situation is more exacerbated for Craft/Technology teachers who have fewer hours per week than mathematics for their subject. Nevertheless, the interviewed students have expressed a preference for hands-on activities taught in the Craft/Technology subject. In the Finnish school involved in the study, much of the time in Craft is dedicated to designing and programming artefacts. This approach was appreciated by students who felt more creative and involved in Craft than Mathematics concerning programming. This was also evident in the focus group interview with students in France, who said they prefer activities in the field of Technology, which were considered more engaging and less demanding (e.g., programming robots) compared to the activities in Mathematics (e.g., developing computational solutions to open problems).

Table 7. Main enablers and barriers in implementing CT within other subjects at lower secondary school

Country	Enablers	Barriers
Finland	Taking advantage of two transition years to accommodate the introduction of programming and algorithmic thinking in the new core curriculum by engaging all relevant stakeholders in the renewal process. Placing CT-related concepts within existing compulsory subjects (Mathematics & Craft) and within specific transversal skills help address gender and equity issues since the development of CT-related concepts is offered to all students. Setting-up support measures for upskilling of teachers.	Address teacher needs for professional development in programming and algorithmic thinking. Lack of clear indications on content, time allocated, and objectives related to programming and algorithmic thinking. The quality of the learning materials and textbooks must be improved. Investments in the infrastructure are needed for teaching algorithmic thinking and programming.
France	A clear policy (e.g., indicating which CT topics/objectives need to be included in the curricula) and an implementation strategy (e.g., including CT-related topics in the national final examinations of the relevant subjects). Prioritizing the implementation at the ISCED 2 level reduced the impact on resources required to support the implementation and integration of programming at schools.	Limited access to technology both at the classroom and at the school level. Students' difficulties with informatics: some students find informatics very difficult and not necessarily students who have difficulty in Mathematics. The time allocated to programming is considered insufficient by the interviewed teachers and students to develop and address these skills adequately.
Sweden	Investing for a long time in the culture of digital competence has favoured the up-take and implementation of programming in schools Having adequate and up-to-date equipment and infrastructure to carry out programming activities.	Adding programming to an already crowded curriculum of Mathematics & Technology. Up-skilling of teachers in developing programming competence.

While the curricula in all three countries do not specify the amount of time for the added CS contents, the mathematics curriculum limits the time teachers can devote to algorithms and programming. The interviewed Mathematics teachers estimated that this time was less than 1 hour per week in all three countries. The situation is more exacerbated for Craft/Technology teachers who have fewer hours per week than mathematics for their subject. Nevertheless, the interviewed students have expressed a preference for hands-on activities taught in the Craft/Technology subject. In the Finnish school involved in the study, much of the time in Craft is dedicated to designing and programming artefacts. This approach was appreciated by students who felt more creative and involved in Craft than Mathematics concerning programming. This was also evident in the focus group interview with students in France, who said they prefer activities in the field of Technology, which were considered more engaging and less demanding (e.g., programming robots) compared to the activities in Mathematics (e.g., developing computational solutions to open problems).

In France, mathematics teachers are encouraged to promote project work and foster collaboration among students in the creative digital production of programs, applications, animations, etc. Use of the Scratch programming environment is recommended as very suitable for this pedagogical approach. In Finland, constructing simple programs generally encourages an active learning approach, particularly in Mathematics. In programming activities, particular attention is placed on generalizing solutions as an introduction to algorithmic thinking. In Technology-Craft, programming is mainly explored through the control of physical objects like robots. In Sweden, the Mathematics curriculum encourages students' use of digital tools and programming to investigate mathematical problems and concepts, make calculations, and interpret data. In lower secondary school, students should gain a basic understanding of programming, dealing with concrete situations.

Regarding tools, it is worth noting that at ISCED 2 level, both the Finnish and Swedish interviewees referred to a shift from visual to text-based programming languages like Python. This shift aims in keeping with the increasing complexity of the topics studied and with the overarching objective of developing students' digital skills for future employment. By contrast, in France a visual-based programming language (namely Scratch) is still used at ISCED 2 level, following ministerial recommendations. The reasoning behind this approach is to continue with an easy-to-use programming environment, thereby reducing potential barriers for teachers and students.

One of the main challenges, as was indicated in the interviews by teachers of Mathematics and Technology, is employing diverse approaches for the integration of CT to be compliant with the teaching practices of the respective disciplines: understanding programming in a more abstract way in Mathematics and implementing more concrete and applicative programming in Technology.

As part of the curriculum reform in France in 2015, Mathematics and Technology teachers offered a two-day compulsory training course organized by the Ministry of Education. In addition, the Ministry's regional offices (called Acadamie), universities and private sector organizations have organized various training courses for teachers in CS. However, France does not require teachers to complete their annual professional training. From the interviews emerged that to address the need for upskilling all Mathematics & Technology teachers, those with more experience promoted peer training opportunities and support in enacting the CT-related curriculum.

Finnish education authorities immediately introduced programming in Mathematics and Crafts classes and launched targeted professional development initiatives to equip teachers with appropriate fundamental skills in programming. The Finnish National Agency for Education (OPH) finances the development and implementation of targeted training courses through regular calls for tender open to public and private organizations and universities. Finnish teachers must fulfil an annual professional development quota of 12-18 hours. The training quota is compulsory for teachers in all subject areas, and a substitute class teacher must be found to fill in these requirements; of course, this causes problems for the school administration. In the view of the OPH expert, those challenges might be overcome through a dedicated tutor teacher, an additional resource in charge of peer tutoring at the school level. This model successfully supported the implementation of transversal competencies in Finland from 2017.

In Sweden, the National Agency for Education has offered online training courses and workshops on programming for Mathematics and Technology teachers. In addition, universities have prepared and launched several courses. Although Swedish teachers have to meet an annual quota of 104 hours of in-service training, these hours are not set to some specific area. So, it could be very different from school to school.

The cluster analysis by word similarities on the transcripts of the interviews and focus groups in MCS3 revealed two main patterns (Figure 4).

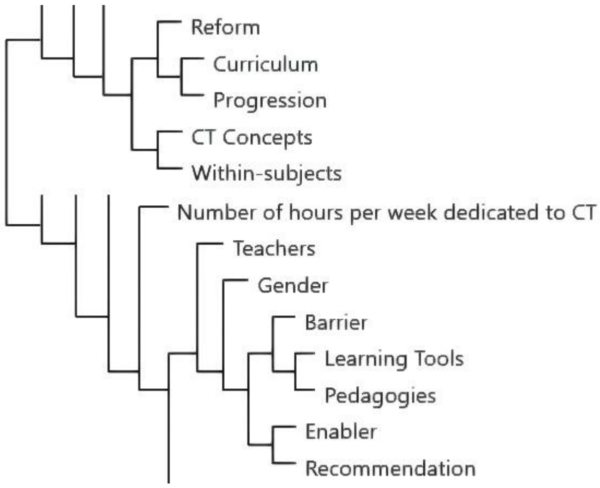

Figure 4. A selection of the cluster analyses hierarchy for CT within other subjects (MCS3)

The first deals with pedagogical approaches to CT integration. The codes “Learning Tools” and “Pedagogies” share a strong correlation (r=.98) with “Pedagogies” connected as well, in the upper branches, with “Barrier” (r=.97), “Progression” (r= .96), “Enabler” (r= .96) and the MCS3 emerged code labelled “Numbers of hours dedicated to CT” (r=.92). Additionally, “Learning” Tools” and “Barrier” emerged as strongly connected (r=.96), as well as “Teachers” and “Gender” (r=.91) and “Enabler” and “Recommendation” (r=.92) thereby suggesting that the pedagogical approach to integrating CT-related contents should link to the subject specificity and has implications on the learning tools adopted to operationalize the different approaches.

The second pattern regards the reforming practices of CT integration. The codes “Within-subjects” and “CT Concepts” emerged as strongly related (r=.93). “CT Concepts” strongly related as well with “Progression” (r=.94), “Curriculum” (r=.95) and “Reform” (r=.93). Finally, “Reform” strongly related to “Curriculum” (r=.95). This points to the relationship between interviewees’ vision of CT and how they position it in terms of students’ digital competence and reform practices.

4. Discussion

4.1. Contributions to Integrating CT into School Curricula

CS concepts supporting the development of CT skills have emerged consistently as part of primary and lower secondary curricula in the nine countries involved in our study. Through integrating basic CS concepts in the curriculum, the respective Ministries of Education aim to address the need to support students with a strong scientific background to reach a better comprehension and solid ability to act in a more complex digital world. The rationale that basic CS concepts underpin core CT skills supporting and complementing digital competence development also emerged from the interviews conducted with policymakers, school leaders, experts, and teachers. Through a comparative analysis of the curricula of the nine countries involved, we identified a common pattern for the basic CS concepts included, which are centred around the relationship between algorithms and programming (Bocconi et al., 2022).

Integrating fundamental CS concepts in the analysed curricula is a relatively recent development, thus competing with more established curriculum priorities. When CS concepts are integrated within existing subjects, they are generally added to an already crowded curriculum. For example, in MCS3, all math teachers complained about the lack of guidelines regarding the time that should be allocated to programming and algorithms and the need to make room for it. Since all curricula examined favour a laboratory approach to programming as a fruitful one to develop computing skills, this implies adequate and recurrent financial support to provide schools with the proper digital infrastructure required.

Finally, supporting teachers in both formative and summative assessments is crucial to improve computing education quality and monitor the evolution of students' CT skills. Primary teachers interviewed in the context of MCS1 pointed out their approach to formative assessment based on observing students while solving problems, developing projects, and learning by reflecting on mistakes. In MCS3, provisions for integrating programming and algorithmic skills in the final examination for Math at the end of ISCED2 were mentioned. It is already part of the Mathematic tests in France, and Sweden plans to move in the same direction. Summative assessment is also crucial for monitoring CT skills development as a foundational component of compulsory education.

4.2. Strengths and Limitations

This study encompasses some limitations. First, it is focused on a small sample of schools, teachers and students from each of the nine European countries. Therefore, the results may not generalize to other schools or education systems. Consequently, future studies should replicate the MCSs across different school settings involving teachers with diverse backgrounds and students from more grades to investigate which results will be consistent with the current research and which will not.

Second, we collected qualitative data through semi-structured interviews and focus groups to shed light on how CT skills are taught and assessed and what tools teachers and learners use. However, more evidence is needed. Future research could collect and analyse more evidence, for instance, through observations, questionnaires, and analysis of teaching and learning materials (e.g., lesson plans, programming artefacts), to gain a deeper understanding of how the school's infrastructure and the different pedagogical approaches affect the findings.

Lastly, 21 out of 40 interviews and 9 out of 10 focus groups were conducted in different languages and were translated (in some cases using machine translation), coded and analysed in English. Although the authors took all possible measures to improve machine translations, some collected data may have been misinterpreted.

4.3. Data Trustworthiness

According to Lincoln and Guba (1985), qualitative results can be evaluated using the "trustworthiness" standard established by credibility and confirmability. The adoption of MCSs and the different data sources (desk research, interviews and focus groups) increased the credibility of the case studies, allowing the development of converging lines of inquiry (Yin, 2018). Replicating the procedures in all nine cases made this study more trustworthy (Yin, 2018). Online team meetings were held regularly throughout the study to establish and clarify research questions, identify cases and participants' selection criteria, and develop interview protocols. Three senior researchers examined the collected data independently and then collectively to reach a consensus about the emerging patterns and the structure of the case reports. Both patterns and case report structure were presented to the rest of the team for comments and validation. Lastly, participants were provided with a temporary copy of the report for the case study they had been involved in to check for internal consistency and coherence of the preliminary findings. This copy also helped strengthen the reliability of the study by ensuring the accuracy of the reported contents (Creswell and Poth, 2018). Overall, interviewees approved the reports and suggested only a few minor adjustments.

4.4. Implications for Future

Introducing basic CS concepts in primary and lower secondary compulsory education is a significant change in the content of the curricula that requires addressing specific challenges in the upskilling of in-service teachers, competing curriculum priorities and appropriate formative and summative assessment measures.

From the three MCS, enablers and barriers emerged in implementing the curricula reforms undertaken in introducing CT. A common enabler is adopting appropriate measures for creating guidelines, learning materials and a large-scale professional development program. On the barrier side, the lack of qualified teachers, lack of quality materials and the challenges related to large-scale upskilling are shared.

In primary education, where teachers are required to cover several subjects in their practice, the number to be involved in teaching CT is quite large. This has implications for the required level of resources to be allocated and the need for developmentally appropriate materials and tools.

In lower secondary education, where there is a clear distinction of subjects, competition among several priorities might impact the successful implementation of computing education. Since most in-service teachers at primary and lower secondary schools do not have a background in computing education, a professional development program is needed to address basic CS concepts and pedagogical approaches appropriate for their student's age.

This large-scale upskilling requires time and support actions. In the MCS, school leaders' support is mentioned among the enablers. UK-England underlines the importance of school hubs connecting schools to support and learn from each other.

Future research could continue to explore CS integration in compulsory education in several directions: i) covering more countries; ii) delving into relevant CS training methods and curricula in action elements that have been tried and tested, iii) focusing deeper on teacher professional development in computing.

Acknowledgments

The authors are grateful to Andreia Inamorato dos Santos from Joint Research Centre of the European Commission (JRC), and to the research fellows Patricia Wastiau, Jeffrey Earp, Milena Horvath, Eglė Jasutė, Vaida Masiulionytė-Dagienė and Gabrielė Stupurienė, for contributing to the design of the multiple-cases studies and various stages of data collection and analysis. The authors would like to extend their special thanks to the research officers from JRC, Romina Cachia and Yves Punie, as well as to the policy officers from DG-EAC, Anusca Ferrari and Veronica Mobilio, for their ongoing support throughout the research process and their helpful feedback on reporting the project results. Finally, we thank the policymakers, experts, school leaders, teachers and students we interviewed for the nine in-depth case studies presented in this paper.

Statements of open data and ethics

All personal information gained about participants and participant identities were fully protected and adequately managed throughout this study in compliance with GDPR provisions. Before and after each interview, partner interviewers assured participants of complete confidentiality. In compliance with GDPR, all the collected data was safely stored on a CNR-ITD server; it will be duly deleted after 5 years.

Gaining informed consent from each participant was a primary ethical concern in this study, as compliance with personal data management provisions was key to GDPR. Therefore, informed consent was given as a precondition for conducting each interview. Involvement in the study was voluntary, and participants were informed from the outset that they had the right to withdraw until the data analysis was completed. For students, consent forms were collected and retained by their respective schools, which were required to sign a declaration stating that both the students and parents had been duly informed and agreed to participate.

Disclaimer

The research presented in this paper is based on the study "Reviewing Computational Thinking in Compulsory Education: State of Play and Practices from the Field" supported by the Joint Research Centre of the European Commission in collaboration with the Directorate General of Education, Youth Sports and Culture under contract no. 941223 - 2021. The information and views set out in this article are those of the authors and do not necessarily reflect the official opinion of the Commission.

References

Arfé, B, Vardanega, T., Ronconi, L. (2020). The effects of coding on children's planning and inhibition skills. Computers & Education 148, (April 2020), 103807. https://doi.org/10.1016/j.compedu.2020.103807

Barendsen, E., Grgurina, N., Tolboom, J. (2016). A New Informatics Curriculum for Secondary Education in The Netherlands. In Informatics in Schools: Improvement of Informatics Knowledge and Perception (Lecture Notes in Computer Science), Springer International Publishing, Cham, 105–117. https://doi.org/10.1007/978-3-319-46747-4_9

Baxter, P., Jack, S. (2008). Qualitative Case Study Methodology: Study Design and Implementation for Novice Researchers. The Qualitative Report 13, 4 (December 2008), 544–559. https://doi.org/10.46743/2160-3715/2008.1573

Bocconi, S., Chioccariello, A., Dettori, G., Ferrari, A., Engelhardt, K. (2016). Developing Computational Thinking in Compulsory Education - Implications for policy and practice. Publications Office of the European Commission. https://doi.org/10.2791/792158

Bocconi, S., Chioccariello, A., Kampylis, P., Dagienė, V., Wastiau, P., Engelhardt, K., Earp, J., Horvath, M.A., Jasutė, E., Malagoli, C., Masiulionytė-Dagienė, V., Stupurienė, G. (2022). Reviewing Computational Thinking in Compulsory Education. Publications Office of the European Union. https://doi.org/10.2760/126955

Braun, V., Clarke, V. (2006). Using thematic analysis in psychology. Qualitative Research in Psychology 3, 2 (January 2006), 77–101. https://doi.org/10.1191/1478088706qp063oa

Creswell, J. W., Guetterman, T. C. (2019). Educational research: planning, conducting, and evaluating quantitative and qualitative research (Sixth edition ed.). Pearson, Saddle River, New Jersey.

Creswell, J. W., Poth, Ch. N. (2018). Qualitative inquiry & research design: choosing among five approaches (Fourth edition ed.). SAGE, Los Angeles.

Csizmadia, A., Standl, B., Waite, J. (2019). Integrating the Constructionist Learning Theory with Computational Thinking Classroom Activities. Informatics in Education 18, 1 (April 2019), 41–67. https://doi.org/10.15388/infedu.2019.03

Curzon, P., Bell, T., Waite, J., Dorling, M. (2019). Computational Thinking. In The Cambridge Handbook of Computing Education Research, Anthony V. Robins and Sally A. Fincher (eds.). Cambridge University Press, Cambridge, 513–546. https://doi.org/10.1017/9781108654555.018

Dagiene, V., Hromkovic, J., Lacher, R. (2022). Designing informatics curriculum for K-12 education: From Concepts to Implementations. Informatics in Education, 20, 3 (September 2021), 333-360. https://doi:10.15388/infedu.2021.22

Dagli, Z., Tokmak, H. S. (2022). Exploring high school computer science course teachers' instructional design processes for improving students' "computational thinking" skills. Journal of Research on Technology in Education 54, 4 (August 2022), 511–534. https://doi.org/10.1080/15391523.2021.1881844

Denning, P.J., Tedre, M. (2022). Computational Thinking: A Disciplinary Perspective. Informatics in Education, 20, 3 (September 2021), 361-390. https://doi:10.15388/infedu.2021.21

Eisenhardt, K. M., Graebner, M. E. (2007). Theory Building From Cases: Opportunities And Challenges. AMJ 50, 1 (February 2007), 25–32. https://doi.org/10.5465/amj.2007.24160888

European Commission. (2020). Digital Education Action Plan 2021-2027 Resetting education and training for the digital age. Retrieved November 9, 2022 from https://eur-lex.europa.eu/legal-content/EN/ALL/?uri=CELEX:52020DC0624

Eurydice. (2019). Digital Education at school in Europe. https://eacea.ec.europa.eu/national-policies/eurydice/sites/default/files/eurydice_brief_digital_education_n.pdf

Fagerlund, J., Häkkinen, P., Vesisenaho, M., Viiri, J. (2020). Assessing 4th Grade Students' Computational Thinking through Scratch Programming Projects. Informatics in Education 19, 4 (December 2020), 611–640. https://doi.org/10.15388/infedu.2020.27

Fessakis, G., Prantsoudi, S. (2019). Computer science teachers' perceptions, beliefs and attitudes on computational thinking in Greece. Informatics in Education. 18, 2 (2019), 227–258. https://doi.org/10.15388/infedu.2019.11

Grgurina, N., Barendsen, E., Suhre, C., Zwaneveld, B., van Veen, K. (2018). Assessment of modeling and simulation in secondary computing science education. In Proceedings of the 13th Workshop in Primary and Secondary Computing Education (WiPSCE '18), Association for Computing Machinery, New York, NY, USA, 1–10. https://doi.org/10.1145/3265757.3265764

Grover, S., Jackiw, N., Lundh, P. (2019). Concepts before coding: non-programming interactives to advance learning of introductory programming concepts in middle school. Computer Science Education 29, 2–3 (July 2019), 106–135. https://doi.org/10.1080/08993408.2019.1568955

Grover, S., Pea, R. (2018). Computational Thinking: A competency whose time has come. In Computer Science Education: Perspectives on Teaching and Learning in School, Sue Sentance, E. Barendsen and S. Carsten (eds.). Bloomsbury, London, UK, 19–38.

Hazzan, O., Ragonis, N., Lapidot, T., Rosenberg-Kima, R. (2020). Computational Thinking. In Guide to Teaching Computer Science. Springer International Publishing, Cham, CH, 57–74. https://doi.org/10.1007/978-3-030-39360-1_4

Hsu, T.-C., Chang, S.-C., Hung, Y.-T. (2018). How to learn and how to teach computational thinking: Suggestions based on a review of the literature. Computers & Education 126, (November 2018), 296–310. https://doi.org/10.1016/j.compedu.2018.07.004

Hsu, Y.-C., Irie, N. R., Ching, Y-H. (2019). Computational Thinking Educational Policy Initiatives (CTEPI) Across the Globe. TechTrends 63, 3 (May 2019), 260–270. https://doi.org/10.1007/s11528-019-00384-4

Huang, W., Looi, C.-K. (202)1. A critical review of literature on "unplugged" pedagogies in K-12 computer science and computational thinking education. Computer Science Education 31, 1 (January 2021), 83–111. https://doi.org/10.1080/08993408.2020.1789411

Israel-Fishelson, R., Hershkovitz, A., Eguíluz, A., Garaizar, P., Guenaga, M. (2021). The Associations Between Computational Thinking and Creativity: The Role of Personal Characteristics. Journal of Educational Computing Research 58, 8 (January 2021), 1415–1447. https://doi.org/10.1177/0735633120940954

Jackson, K., Bazeley, P. (2019). Qualitative data analysis with NVivo (3rd edition ed.). SAGE Publications, Thousand Oaks, CA.

Jocius, R., Joshi, D., Dong, Y., Robinson, R., Cateté, V., Barnes, T., Albert, J., Andrews, A., Lytle, A. (2020). Code, Connect, Create: The 3C Professional Development Model to Support Computational Thinking Infusion. In Proceedings of the 51st ACM Technical Symposium on Computer Science Education (SIGCSE '20), Association for Computing Machinery, New York, NY, USA, 971–977. https://doi.org/10.1145/3328778.3366797

Kale, U., Akcaoglu, M., Cullen, T., Goh, D., Devine, L., Calvert, N., Grise, K. (2018). Computational What? Relating Computational Thinking to Teaching. TechTrends 62, 6 (November 2018), 574–584. https://doi.org/10.1007/s11528-018-0290-9

Kert, S. B., Kalelioğlu, F., Gülbahar, Y. (2019). A Holistic Approach for Computer Science Education in Secondary Schools. Informatics in Education 18, 1 (April 2019), 131–150. https://doi.org/10.15388/infedu.2019.06

Kong, S.-C. (2016). A framework of curriculum design for computational thinking development in K-12 education. J. Comput. Educ. 3, 4 (December 2016), 377–394. https://doi.org/10.1007/s40692-016-0076-z

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

Kong, S.-C., Lai, M., Sun, D. (2020). Teacher development in computational thinking: Design and learning outcomes of programming concepts, practices and pedagogy. Computers & Education 151, (July 2020), 103872. https://doi.org/10.1016/j.compedu.2020.103872

Lai, X., Wong, G. K-w. (2022). Collaborative versus individual problem solving in computational thinking through programming: A meta-analysis. British Journal of Educational Technology 53, 1 (2022), 150–170. https://doi.org/10.1111/bjet.13157

Lee, S. J. Francom, G. M., Nuatomue, J. (2022). Computer science education and K-12 students' computational thinking: A systematic review. International Journal of Educational Research 114, (January 2022), 102008. https://doi.org/10.1016/j.ijer.2022.102008

Lincoln, Y. S., Guba, E. G. (1985). Naturalistic inquiry. Sage Publications, Beverly Hills, Calif.

Litts, B. K., Searle, K. A., Brayboy, B. M. J., Kafai, Y. B. (2021). Computing for all?: Examining critical biases in computational tools for learning. British Journal of Educational Technology 52, 2 (2021), 842–857. https://doi.org/10.1111/bjet.13059

Manches, A., Plowman, L. (2017). Computing education in children's early years: A call for debate. British Journal of Educational Technology 48, 1 (2017), 191–201. https://doi.org/10.1111/bjet.12355

Metcalf, S. J., Reilly, J. M., Jeon, S., Wang, A., Pyers, A., Brennan, K., Dede C. (2021). Assessing computational thinking through the lenses of functionality and computational fluency. Computer Science Education 31, 2 (April 2021), 199–223. https://doi.org/10.1080/08993408.2020.1866932

Niemelä, P., Partanen, T., Harsu, M., Leppänen, L., Ihantola, P. (2017). Computational thinking as an emergent learning trajectory of mathematics. In Proceedings of the 17th Koli Calling International Conference on Computing Education Research (Koli Calling '17), Association for Computing Machinery, New York, NY, USA, 70–79. https://doi.org/10.1145/3141880.3141885

del Olmo-Muñoz, J., Cózar-Gutiérrez, R., González-Calero, J. A. (2020). Computational thinking through unplugged activities in early years of Primary Education. Computers & Education 150, (June 2020), 103832. https://doi.org/10.1016/j.compedu.2020.103832

Palts, T., Pedast, M. (2020). A Model for Developing Computational Thinking Skills. Informatics in Education 19, 1 (March 2020), 113–128. https://doi.org/10.15388/infedu.2020.06

Paniagua, A., Istance, D. (2018). Teachers as Designers of Learning Environments - The Importance of Innovative Pedagogies. OECD Publishing, Paris.

Perin, A. P. J., dos S. Silva, D. E., Valentim, N. M. C. (2023). Investigating block programming tools in high school to support Education 4.0: A Systematic Mapping Study. Informatics in Education, 22, 3 (September 2023), 463-498. https://doi:10.15388/infedu.2023.21

Piedade, J., Dorotea, N. (2023). Effects of Scratch-based activities on 4th-grade students' computational thinking skills. Informatics in Education, 22, 3 (September 2023), 499-523. https://doi:10.15388/infedu.2023.19

Polat, E., Hopcan, S., Kucuk, S., Sisman, B. (2021). A comprehensive assessment of secondary school students' computational thinking skills. British Journal of Educational Technology 52, 5 (2021), 1965–1980. https://doi.org/10.1111/bjet.13092

Repenning, A., Basawapatna, A., Escherle, N. (2016). Computational thinking tools. In 2016 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC), IEEE, Cambridge, United Kingdom, 218–222. https://doi.org/10.1109/VLHCC.2016.7739688

Román-González, M., Moreno-León, J., Robles, G. (2019). Combining Assessment Tools for a Comprehensive Evaluation of Computational Thinking Interventions. In Computational Thinking Education, Siu-Cheung Kong and Harold Abelson (eds.). Springer Open, Singapore, 79–98.

Román-González, M., Pérez-González, J.-C., Jiménez-Fernández, C. (2017). Which cognitive abilities underlie computational thinking? Criterion validity of the Computational Thinking Test. Computers in Human Behavior 72, (July 2017), 678–691. https://doi.org/10.1016/j.chb.2016.08.047

Ryan, G. W., Bernard, H. R. (2003). Techniques to Identify Themes. Field Methods 15, 1 (February 2003), 85–109. https://doi.org/10.1177/1525822X02239569
Sáez-López, J.-M., Román-González, M., Vázquez-Cano, E. (2016). Visual programming languages integrated across the curriculum in elementary school: A two year case study using "Scratch" in five schools. Computers & Education 97, (June 2016), 129–141. https://doi.org/10.1016/j.compedu.2016.03.003
Sherwood, H., Yan, W., Liu, R., Martin, W., Adair, A., Fancsali, C., Rivera-Cash, E., Pierce, M., Israel, M. (2021). Diverse Approaches to School-wide Computational Thinking Integration at the Elementary Grades: A Cross-case Analysis. In Proceedings of the 52nd ACM Technical Symposium on Computer Science Education (SIGCSE '21), Association for Computing Machinery, New York, NY, USA, 253–259. https://doi.org/10.1145/3408877.3432379
Statter, D., Armoni, M. (2020). Teaching Abstraction in Computer Science to 7th Grade Students. ACM Trans. Comput. Educ. 20, 1 (Gennaio 2020), 8:1-8:37. https://doi.org/10.1145/3372143
Syslo, M. (2020). The new computer science curriculum in Poland - challenges and solutions. In K. Kori and M. Laampere (Eds.), Proceedings of the International Conference on Informatics in School: Situation, Evaluation and Perspectives (pp. 120–127). http://ceur-ws.org/Vol-2755/paper10.pdf
Taslibeyaz, E., Kursun, E., Karaman, S. (2020). How to Develop Computational Thinking: A Systematic Review of Empirical Studies. Informatics in Education 19, 4 (December 2020), 701–719. https://doi.org/10.15388/infedu.2020.30
Tikva, C., Tambouris, E. (2021). Mapping computational thinking through programming in K-12 education: A conceptual model based on a systematic literature Review. Computers & Education 162, (March 2021), 104083. https://doi.org/10.1016/j.compedu.2020.104083
Tonbuloğlu, B., Tonbuloğlu, I. (2019). The Effect of Unplugged Coding Activities on Computational Thinking Skills of Middle School Students. Informatics in Education 18, 2 (October 2019), 403–426. https://doi.org/10.15388/infedu.2019.19
Tsai, M.-J., Liang, J.-C., Hsu, C.-Y. (2021). The Computational Thinking Scale for Computer Literacy Education. Journal of Educational Computing Research 59, 4 (July 2021), 579–602. https://doi.org/10.1177/0735633120972356
UNESCO Institute for Statistics. (2012). International Standard Classification of Education (ISCED) 2011. UNESCO Institute for Statistics. https://doi.org/10.15220/978-92-9189-123-8-en
Upadhyaya, B., McGill, M. M., Decker, A. (2020). A Longitudinal Analysis of K-12 Computing Education Research in the United States: Implications and Recommendations for Change. In Proceedings of the 51st ACM Technical Symposium on Computer Science Education (SIGCSE '20), Association for Computing Machinery, New York, NY, USA, 605–611. https://doi.org/10.1145/3328778.3366809
Webb, M., Davis, N., Bell, T., Katz, Y., Reynolds, N., Chambers, D. P., Sysło, M. M. (2017). Computer science in K-12 school curricula of the 2lst century: Why, what and when? Educ. Inf. Technol. 22, 2 (2017), 445–468. https://doi.org/10.1007/s10639-016-9493-x
Wei, X., Lin, L., Meng, N., Tan, W., Kong, S.-C., Kinshuk. (2021). The effectiveness of partial pair programming on elementary school students' Computational Thinking skills and self-efficacy. Computers & Education 160, (January 2021), 104023. https://doi.org/10.1016/j.compedu.2020.104023
Yadav, A., Gretter, S., Hambrusch, S., Sands, P. (2016). Expanding computer science education in schools: understanding teacher experiences and challenges. Computer Science Education 26, 4 (December 2016), 235–254. https://doi.org/10.1080/08993408.2016.1257418
Yağcı, M. (2019). A valid and reliable tool for examining computational thinking skills. Educ Inf Technol 24, 1 (January 2019), 929–951. https://doi.org/10.1007/s10639-018-9801-8
Yin, R. K. (2018). Case study research and applications: design and methods (Sixth edition ed.). SAGE, Los Angeles.

Zach, L. (2006). Using a Multiple-Case Studies Design to Investigate the Information-Seeking Behavior of Arts Administrators. Library Trends 55, 1 (September 2006), 4–21. https://doi.org/10.1353/lib.2006.0055

Zhang, L., Nouri, J. (2019). A systematic review of learning computational thinking through Scratch in K-9. Computers & Education 141, (November 2019), 103607. https://doi.org/10.1016/j.compedu.2019.10360

Appendix. Semi-structure interview

A. MCS expert interview questions

For MCS experts, we include both the general questions and the adaptations that take into account the specificity of each MCS (i.e., CT integrated as (a) a cross-curriculum topics; (b) as a separate subject; (c) within other subjects).

A.0. General

1. What is, in your view/experience, the relationship between CT and CS/Informatics/Computing?
2. Since the debate on the definition of CT in still ongoing, which set of key concepts and practices should characterized CT education? How this impact on teaching and assessment?
3. What is the underlying rationale for introducing CT in compulsory education?
4. Why should CT be introduced in compulsory education as (a separate subject / within existing disciplines / across disciplines)?
5. What are the benefits and challenges of introducing CT as education as (a separate subject / within existing disciplines / across disciplines) in compulsory education?
6. How CT curriculum should be organised in compulsory education concerning:
   - Age appropriateness
   - Pedagogy
   - Gender
   - Equity
7. What is the role of programming in CT curriculum in compulsory education?
8. For multiple-case study X, we will analyse the implementation of CT/I/CS/C as (a separate subject / within existing disciplines / across disciplines) at (lower secondary/primary) level in three countries (i.e., XXX, YYY and ZZZ). In your opinion, when comparing these different educational systems, to what we should pay particular attention?
9. How teacher education and professional development should be organised in relation to introducing CT in compulsory education?

What is your main recommendation to policy makers in Europe for Quality Computing Education?

A1. MCS1 specific

CT understanding in compulsory education

1. What is the relationship between Computational Thinking and Computer Science/Informatics/Computing?
2. Since the debate on the definition of Computational Thinking in still ongoing, which set of key concepts and practices should characterize CT in education? How this impact on teaching and assessment?

CT across disciplines in primary education

3. What is the underlying rationale for introducing CT across disciplines in primary education?
4. Why should CT be introduced across disciplines in primary education?
5. What are the benefits and challenges of introducing CT across disciplines in primary education?
6. How CT in Math curriculum should be organised in primary education concerning:

  - Age appropriateness
  - Pedagogy
  - Gender
  - Equity
7. What is the role of programming in primary education?

CT assessment and professional development

8. How should learning of CT concepts and skills be assessed in primary education?
9. How teacher education and professional development should be organised in relation to introducing CT in primary education?

Other

10. For MCS1, we will analyse CT implementation across disciplines in primary school in three countries (i.e., Lithuania, Norway, and Slovakia). In your opinion, when comparing these different educational systems, to what we should pay particular attention?

What is your main recommendation to policy makers in Europe for Quality Computing Education?

A2. MCS2-specific

CT understanding in compulsory education

1. What is, in your view/experience, the relationship between CT and CS / Informatics/Computing?
2. Since the debate on the definition of CT in still ongoing, which set of key concepts and practices should characterize CT in education? How does this impact on teaching and assessment?

CT as a separate subject in lower secondary education

3. What is the underlying rationale for introducing Informatics into compulsory education?
4. Why should Informatics be introduced in compulsory education as a separate subject?
5. What are the benefits and challenges of introducing Informatics as a separate subject in compulsory education?
6. How should the Informatics curriculum be organised in compulsory education concerning:
   - Age appropriateness
   - Pedagogy
   - Gender
   - Equity
7. What is the role of programming in the Informatics curriculum in compulsory education?

CT assessment and professional development

8. How should learning of CT concepts and skills be assessed in lower secondary education?
9. How should teacher education and professional development be organised in relation to introducing Informatics in compulsory education?

Other

10. For the MCS 2, we will analyse the implementation of Informatics as a separate subject at lower secondary level in three countries (i.e., Croatia, Poland and England). In your opinion, when comparing these different educational systems, to what should we pay particular attention?

What is your main recommendation to policy makers in Europe for Quality CS Education?

A3. MCS3-specific

CT understanding in compulsory education

1. What is the relationship between CT and CS/Informatics/Computing?
2. Since the debate on the definition of CT in still ongoing, which set of key concepts and practices should characterized CT education? How this impact on teaching and assessment?

CT within existing disciplines (e.g., Math) in lower secondary

3. What is the underlying rationale for introducing CT in Math curriculum in compulsory

education?
4. Why should CT be introduced in compulsory education within existing disciplines e.g., Math?
5. What are the benefits and challenges of introducing CT within existing disciplines in lower secondary education?
6. How CT in Math curriculum should be organised in lower secondary education concerning:
   - Age appropriateness
   - Pedagogy
   - Gender
   - Equity
7. What is the role of programming in Math curriculum in lower secondary education?

CT assessment and professional development

8. How should learning of CT concepts and skills in Math be assessed?
9. How teacher education and professional development should be organised in relation to introducing CT in compulsory education?

Other

10. For MCS 3, we will analyse the implementation of CT within existing disciplines (i.e., Math, Tech) at lower secondary level in three countries (i.e., France, Finland and Sweden). In your opinion, when comparing these different educational systems, to what we should pay particular attention?
11. One dimension of CT is its potential to foster new ideas and approaches in science (e.g., computational dimensions are present in a great variety of domain, such as computational biology, computational geography. How could this dimension of CT be exploited in the context of compulsory education?

What is your main recommendation to policy makers in Europe for Quality CS Education?

B. Policy maker interview questions

B1. Definition and conceptualisation of CT skills

1. How is CT defined in the curriculum and in practice? What terminology is used in relation to CT in the curriculum?
2. What is, in your view/experience, the relationship between CT [or the term you use in your country] and Computer Science/Informatics?

B2. The implementation of the CT/programming-related activities in the curriculum

3. In year [XXXX], CT/CS/I/C became part of your curriculum. After several years of implementation, what is the status of this new curriculum?
4. Why was CT/programming included in the curriculum as [general cross-curricular theme/ as part of separate subject, within other subject(s),…] (i.e., following what rationale)?
5. How much time has been allocated to the CT/programming activities in the curriculum?
6. How are gender balance and equity considered in implementing the curriculum regarding CT/programming activities?
7. Has the implementation of the curriculum that integrates CT/programming activities been monitored in any way?
8. Have the materials developed for CT/programming activities proved effective in supporting implementation? Is there any plan for updating/revising such materials?
9. What technologies are generally employed?
10. Are other stakeholders (grassroots, university networks, consortia, industry) involved in the implementation of the curriculum?

B3. CPD implementation and approaches

11. Are new teachers being recruited to cover the curriculum as regards CT/programming activities?
12. How is teachers' upskilling being addressed?
13. What methods are being used (e.g., active learning, tutorials)?
14. Who carries out the CPD activities?

B4. Assessment of CT/programming activities

15. How is learning of CT and CS/I/C concepts and skills assessed?
16. Have CT and CS/I/C assessment methods been reviewed and adapted, and if so, how?

B5. CT/programming in iVET education

17. To what extent are CT and CS/I/C concepts included in all iVET tracks (within the limit of compulsory education)? Or are CT and related concepts integrated only in some specific iVET tracks?
18. How can CT and CS/I/C be integrated into different iVET models adopted in compulsory education, i.e., work-based, school-based or dual-VET systems?
19. How can policy makers and different stakeholders (i.e., experts, industry, grassroots) support integration of CT and related topics in compulsory education iVET

What is your main recommendation to policy makers in Europe for Quality CS Education?

C. School leader interview questions

Background information on curriculum renewal in compulsory education.
In year xxx, CT/CS/I/C was introduced in xxxx Curriculum.

1. When did you start integrating CT/CS/I/C in your school?
2. The European Commission uses the term "computational thinking". Did you use this term in your school? If not, which terminology do you use in school?

CT/CS/I/C Curriculum Implementation at school level

3. Has CT/CS/I/C been included in the strategic plan of your school?
4. How synergies and integration of CT/CS/I/C across/within subjects has been supported for a quality informatics education? (USE ONLY FOR MULTIPLE CASE 1 & 3)
5. To what extent teachers in your school integrate CT/CS/I/C in their teaching?
6. What kind of support the school has provided to teachers for integrating CT/CS/I/C in their subjects?
7. How has the school community (teachers, parents) been informed about this change in the curriculum? Have external actors (grassroots, associations) been involved in the innovation of the curriculum? If yes, how?
8. Does the school have the necessary technological infrastructure to integrate CT/CS/I/C in the curriculum?
9. How are gender balance and equity in CT/CS/I/C taken into account at your school?
10. How have assessment methods been reviewed and adapted in relation to CT/CS/I/C?
11. What type of professional development the school offered to teachers on CT/CS/I/C?

What is your main recommendation to your policy makers for improving Quality CS education?

D. Teacher interview questions

Background information on curriculum renewal in compulsory education.
In year xxx, CT/CS/I/C was introduced in Curriculum as a separate subject xxx/ cross curriculum topic in Primary /lower secondary education

1. When did you start integrating CT/CS/I/C in your teaching?
2. The European Commission uses the term "computational thinking". Did you use this term in your teaching practice? If not, which terminology do you use in school?

D1. CPD activities on CT/CS/I/C

3. Did you already know key concepts and related pedagogical approaches to integrate CT/CS/I/C in your teaching?
4. What type of professional development activities provided you with necessary skills?
5. What methods were used (e.g., active learning, tutorials)?
6. Who carried out the CPD activities?

D2. Curriculum Implementation (teaching & assessment)

7. In your national curriculum /curriculum guidelines, the following concepts, related skills and tools (in the table) are indicated. Which adaptations/integrations did you apply?

	ADD HERE SUBJECT NAME FROM THE CURRICULUM
Knowledge	
Related skills	
Tool	

8. How do you teach CT/CS/I/C concepts and related skills in an age-appropriate way?
9. What pedagogical approach do you use?
10. What technologies do you use?
11. How much time do you actually devote to address CT/CS/I/C in the curriculum?
12. Do you use the materials developed by the MOE for CT/CS/I/C?
13. How are gender balance and equity taken into account in your teaching?
14. How do you assess students' learning of CT/CS/I/C concepts and skills in your subject?

What is your main recommendation to your policy makers for Improving Quality CS education?

E. Student Focus group questions

1. Which term do you normally use with other students and teachers when talking about CT/CS/I/C in classroom?

Additional question if needed: Do you know the term computational thinking?

2. What activities related to CT/CS/I/C are proposed in classroom?

Examples if needed: (Math) e.g., develop a program that uses variables, define an algorithm to solve a problem; (Tech) e.g., define instructions for a robot to avoid an obstacle

3. What do you think about these activities (e.g., easy/difficult, fun or not)? Which ones do you think are best and why?
4. How many lessons are dedicated to CT/CS/I/C in your school (Many, quite a few, some, not many, only 1 or 2)? Do you think they are sufficient?
5. What do you think is the purpose of learning about CT/CS/I/C at school?
6. What are your suggestions to improve CT/CS/I/C education?

Received June 13, 2024, accepted June 23, 2024