International Journal of Technology and Design Education
https://doi.org/10.1007/s10798-023-09860-8

Factors influencing Swedish grades 4–6 technology teachers' choice of teaching and learning material in programming education

Eva-Lena Bjursten[1] · **Tor Nilsson**[1] · **Gunnar Jonsson**[2]

Accepted: 27 September 2023

Abstract
There is a recognized need to understand the current state of programming implementation in the Swedish compulsory school system. This study focused specifically on the implementation of programming in the school subject of technology for grades 4–6. In Sweden, the responsibility for choosing teaching and learning material lies with individual teachers. Recent studies have indicated the prevalence of visual programming languages (VPLs) in classrooms. However, no empirical research has specifically investigated why teachers select particular programming learning environments (PLEs) and the challenges they have overcome in this process. Therefore, this study aimed to explore the PLEs used by teachers and the factors influencing their choices. In addition, this study explored the role of pedagogical content knowledge (PCK) and the influence of systemic and situational amplifiers and filters in shaping the programming education landscape, highlighting the importance of understanding these factors for effective implementation. Semi-structured interviews were conducted with 14 experienced programming teachers in grades 4–6 to gather insights. The results revealed that VPLs, particularly Scratch, have been widely adopted, but the study also identified three textual programming languages being utilized. Furthermore, the findings indicate that teachers' previous education plays a significant role in shaping their PLE preferences. This suggests that programming education in both professional development and preservice teacher training is crucial for effective implementation. By investigating PLE choices and the factors influencing them, this study contributes to a better understanding of the current landscape of programming education in Sweden's compulsory school system.

Keywords Pedagogical content knowledge (PCK) · Swedish compulsory school grades 4–6 · Technology · Teacher choice · Programming learning environment (PLE)

Introduction

Changes in society often result in changes in curriculum content. The European Parliament, in its decision to incorporate digital competence into lifelong learning (EU, 2006), marked a significant turning point. While several countries had made attempts

Extended author information available on the last page of the article

Published online: 24 October 2023

to integrate computer programming into their curricula before the EU's decision, it was not until 2006 that the momentum truly caught up with us. This pivotal moment prompted countries across Europe to embrace computer programming as an integral part of their educational programs. In England, this resulted in a new subject—computing—with specific instructions as to what should be taught (e.g. using sequence, selection and repetition in programs, working with variables and understanding key algorithms; (the Office for Standards in Education, 2013). Many of these programming concepts are seen as essential in programming education (Saeli et al. 2010, 2012). In Sweden, however, computer programming was combined with two existing subjects, mathematics and technology, in which the content is described broadly and without specific instructions on what should be taught (Statens Skolverk, 2018). (Hereafter, the subject names are indicated with *italics*.) Therefore, it is up to each Swedish teacher to determine how to include programming in *technology* and in what ways. In this article, programming learning environment (PLE) refers to a specially designed digital platform where a beginner can learn to program (Kelleher & Pausch, 2005). PLEs can be either open [flexible and allowing creativity and experimentation, e.g. self-contained programming; Rich et al., (2022)] or closed [predetermined tasks, e.g. software/game development kits or learning games; Rich et al. (2022)].

Additionally, in Sweden, teachers are free to choose suitable teaching and learning material (TLM). However, few studies have examined how teachers choose TLM. Reichenberg (2014) concluded that content, previous experience and other colleagues' opinions influence the choice of TLM but that most teachers tend to choose what their colleagues recommend. Consequently, as visual programming languages (VPLs), or block programming, and tangible robots have been found popular in previous studies (Humble, 2021), this may affect teachers' choices. Nevertheless, Humble's data were collected from an open forum, which may not accurately reflect the use of PLEs in Swedish schools.

When new content is added to the curriculum, it can lead to major changes in teachers' practices, and any such change can cause teachers difficulties (Ertmer & Ottenbreit-Leftwich, 2010). Correspondingly, it is essential to consider the various individual, situational and systemic A & F that may have an effect. It is important to note that while these factors have the potential to influence teachers' practices, they may not be the sole determinant of their actions; rather, they can be seen as a way to explain the contextual factors that may affect a teacher's idealistic classroom intentions (Doyle et al., 2019). For example, when *technology* was implemented in the curriculum for primary schools in Queensland, Australia, teachers' intrinsic and extrinsic challenges were investigated and strategies to prevent them were developed (Finger & Houguet, 2009). In computer programming, it appears that teachers confront a multitude of obstacles, such as feeling cut off, a lack of appropriate understanding of programming and an absence of professional education and other resources (Yadav et al., 2016). Recent evidence suggests that these challenges also occurred in Sweden when computer programming was added to the curriculum (Vinnervik, 2020). However, it is not yet clear what challenges teachers who have included computer programming have managed to overcome.

This paper aims to characterize the TLM choices of *technology* teachers in grades 4–6 (pupils aged 10–12) who have taken on the teaching of programming. The research questions were as follows:

1. What TLM do Swedish teachers in grades 4–6 include in *technology*?
2. What factors influenced their choice of TLM?

Background

Programming in Swedish curricula

This is not the first time that programming has been included in Swedish curricula. For a short period in the 1980s, a new subject called datalära (i.e. computer science) was included in the curriculum, but was removed from subsequent curriculum due to the lack of software and trained teachers (Utbildningsdepartementet, 2002; Vinnervik, 2021). When programming was reintroduced into the curriculum in 2018 as part of strengthening pupils' digital literacy, the new core content was introduced mainly in *mathematics* and *technology* (Statens Skolverk, 2018). Unquestionably, this new subject content created several challenges for teachers.

First, the description of programming in *technology* is broad in the syllabus: 'Controlling pupils' own constructions or other objects by means of programming' (Statens Skolverk, 2018, p. 298). Furthermore, in support material concerning digitization in Swedish schools, programming is described not just as writing code but also as a means of solving problems (i.e. formulating the problem, choosing a solution and testing, debugging and documenting it). Additionally, official guidance stated that pupils should be given opportunities to see programming from a broader perspective through creative creation, control and regulation, simulation and democratic dimensions (Statens Skolverk, 2017a). In contrast, common concepts are not visible in the Swedish curricula, in contrast to how computer programming was included in the computing syllabus in England (the Office for Standards in Education, 2013).

Second, in the *mathematics* syllabus, the recommendation is to use a VPL, which offers a graphical approach to programming with pre-programmed icons. To construct a program, the programmer moves the icons and glues them together with an easy drag-and-drop technique. Many VPLs include an option to configure the icons easily. Comparatively, to construct a program in a textual programming language (TPL), the programmer has to write the program using the correct spelling, syntax and special keys like ()"}*. Moreover, a TPL is often case-sensitive. Existing research recognizes VPLs as suitable for novice programmers because they make it easy and quick to create a program (Price & Barnes, 2015; Weintrop & Wilensky, 2015). However, questions have been raised about the use of VPLs. Since Swedish pupils most likely will have to learn a TPL later (Statens Skolverk, 2017b), it may be better to learn it from the beginning (Meerbaum-Salant et al., 2011). Not only are pupils novices, but over half of teachers recognize the need to improve their skills to equip their pupils with effective digital literacy (Statens Skolverk, 2019).

Teachers can use knowledge requirements in the curriculum to develop lesson plans focused on meeting learning objectives. Assessment can help identify areas requiring more attention, and feedback from pupils can provide insight into the topics that are most significant to them, allowing teachers to adjust their teaching methods accordingly. However, when computer programming was introduced into the curricula, no knowledge requirements were changed. Since there are no obvious knowledge requirements in the Swedish curricula that can be used to assess pupils' programming projects, this creates uncertainty for teachers, and their assessment is based on other elements of the pupils' work other than the code or how the program solved the problem (Vinnervik, 2021).

Teaching and learning materials in computer programming

In Sweden, there is no system for approving TLM (Reichenberg, 2014), and few studies have examined how teachers choose TLMs. In a study of 319 Swedish teachers, Reichenberg (2014) found that most teachers responded that the most important factor when choosing TLM was content, followed by past experience and collegial recommendations (p. 86). By talking to colleagues, teachers increased the likelihood that they would choose textbooks recommended by their colleagues. Moreover, the longer a teacher had been working, the less important the content was in the choice of the TLM. In this case, however, the question was how the teachers chose textbooks and traditional TLM, such as books and slides. Admittedly, traditional TLM is not appropriate for programming teaching; furthermore, traditional TLM is static and does not provide immediate feedback (Cheah, 2020; Zhang et al., 2013). Nevertheless, while studying teachers' choices, the above factors may be relevant to programming education.

Physical objects are a common tool in primary programming (Rich et al., 2022). A programmable robot is a tangible object that can be programmed to perform actions. Through, the use of a robot, children can learn programming concepts, including input, output, sequencing, loops and conditional statements (Przybylla & Romeike, 2015). However, the task of choosing the right educational robot has become increasingly complex in recent years, primarily due to the integration of advanced features and capabilities in modern robots (Evripidou et al., 2022). When selecting an educational robot, it is essential to consider not only technical aspects such as the type of robot, the number of motors and sensors, programming options (e.g., physical interaction, VPL, or TPL), and compatibility with computing devices, but also pedagogical factors. These pedagogical considerations encompass what pupils are expected to achieve using the robot (Evripidou et al., 2022). For example, when comparing learning gains in programming with a tangible (physical) or a simulated (virtual) object, on the one hand, Fessard et al., (2019) demonstrated that both methods resulted in significant learning gains in programming. However, they found no significant difference between the two methods in terms of learning conditional and iterative structures.

TLM in computer programming in Sweden has been studied before. Humble (2021) described in four groups the programming tools listed on a Swedish website for K–12 teachers (www.Lektion.se), where teachers can share lesson plans:

- Tangible programming involves a physical programmable object;
- Textual programming;
- Block programming or VPLs; and
- Unplugged programming is a style of coding using pictures and/or arrows to 'program' each other to do something [i.e. pupils can learn coding concepts without using a digital device; (Bell & Vahrenhold, 2018; Kong et al., 2020)].

Humble (2021) found that VPLs are common in all compulsory school grades. Of 26 analysed lesson plans, 7 referred to VPLs, of which 3 referred to Scratch, which has become increasingly popular in recent years. By comparison, textual programming was not present in grades 1–6. Furthermore, 11 lessons were directed to education with tangible robots, of which 7 were aimed at the Blue-Bot®. However, as www.Lektion.se is an open forum, the content may not reflect the extent to which each PLE is used in schools in Sweden. Moreover, as there is a plethora of PLEs, it can be difficult for teachers to

choose the most appropriate one for their pupils. In another study on trends in tools for teaching computational thinking, Rich et al. (2022) categorized more than 300 coding tools into 3 categories and 10 subcategories. The software category, includes subcategories such as self-contained programming environments (e.g. the open PLE, Scratch), software/game development kits and learning games. The second category, hardware, includes subcategories such as tangible computing, robotics and microcontrollers and microprocessors, which includes devices that control physical elements such as motors or LEDs (e.g. micro:bit). The third category, unplugged, has four subcategories: games, which refers to board or card games that focus on teaching computational concepts, as well as books, hands-on manipulative and other, which includes any unique tools not already listed that can help young pupils learn coding concepts through coding to interact with these tools. One example of a tool in this category is CT Illustrations. However, the study did not distinguish between VPLs and TPLs, which is an interesting question regarding which type of programming language is suitable for which pupil level. One potential solution to this problem has been presented by Fessakis et al. (2019), who proposed a classification system with five axes to make it easier to choose tools depending on the teaching goals:

- *Axis A* PLE Programming language. This axis concerns the level of abstraction of the programming language. Here we find a span of programming languages from low-level programming languages, which are programmed with complex statements and can allocate memory space, to high-level programming languages, which have a more human-friendly syntax, and further on to even higher-level programming languages, for example, VPLs with their pre-programmed graphical objects.
- *Axis B* Suitability for the pupils' developmental level. Fessakis et al. (2019) argued that this can often be dictated by pupils' general skills.
- *Axis C* How many programming models the PLE supports. For example, event-driven (where functions are triggered by being called after an incoming event), visual (with graphical pre-programmed objects) and procedural programming models (where code is read from left to right and from top to bottom) are three common programming models in primary schools.
- *Axis D* Number of supported programming languages—the more the better.
- *Axis E* Abstraction level of the programming process. The level of abstraction influences pupils' perceptions of the meaning of programming.

Challenges in technology education

In examining the reintroduction of programming in Swedish schools, Vinnervik (2020) studied the challenges teachers faced due to the new content four months before programming was included in the Swedish curriculum. The study involved 19 teachers teaching grades 1–9. The data collection consisted of focus group discussions in March 2018. Few participating teachers in his study had experience in programming, either as programmers or as teachers. Vinnervik did not participate in the study focus groups; instead, the teachers described their individual experiences and challenges and then discussed digitization and programming with each other. In his discussion, Vinnervik notes that the teachers wanted to integrate programming into their teaching and considered it relevant. However, some teachers raised concerns about activities lacking a purpose or goal and focusing on entertainment value. The Swedish Schools Inspectorate has previously highlighted this issue as

a potential problem, as criticism has been directed towards the *technology* subject for the unreflected use of ready-made teaching materials (Skolinspektionen, 2014). Vinnervik's (2020) results indicate that teachers desire teaching materials that can help them deliver what they perceive is expected of them. In addition, guidelines can assist with the successful integration of digital devices in the classroom (Makki et al., 2018). Nevertheless, Vinnervik (2020) recommends assessing learning material quality before purchase as it varies. One of the shortcomings reported by Vinnervik's participants was time, and alongside resource allocation, issues related to professional development and support emerged as particularly pressing concerns that required discussion. This view is supported by Yadav et al. (2016), who wrote that teachers face both content and pedagogical challenges of teaching computer science, emphasizing the need to prepare teachers through teacher education programmes. According to Vinnervik (2020), Swedish teachers expect their school leaders to take control of the situation to meet the need for professional development, but the conditions for professional development vary both between and within schools. This expectation is in line with the importance of school leaders as key actors in the implementation of reforms, as emphasized by Statens Skolverk (2018).

Together, the above-mentioned studies emphasize the need for methods to overcome these barriers and present suggestions on how to counteract the barriers that may arise. Both pedagogical and content challenges exist (Yadav et al., 2016), and teachers must be given time to build the subject-specific teaching skills (Ertmer & Ottenbreit-Leftwich, 2010) that are necessary for good teaching (Vinnervik, 2020). Through various meeting places, teachers can be offered opportunities to discuss and reflect on the subject content and share successful examples (Ertmer & Ottenbreit-Leftwich, 2010; Finger & Houguet, 2009; Yadav et al., 2016). In addition, teachers require time for professional development, such as courses, meetings and workshops, to familiarize themselves with the content and experiment with it (Ertmer & Ottenbreit-Leftwich, 2010; Finger & Houguet, 2009). In addition, these professional development events are not only needed once, but are required on multiple occasions (Yadav et al., 2016) and within regular working hours (Vinnervik, 2020). Ertmer and Ottenbreit-Leftwich (2010) and Finger and Houguet (2009) suggested that networks for teachers and the resources purchased by schools could perhaps be shared between schools (Finger & Houguet, 2009). However, there are risks that teaching will be planned according to purchased PLEs rather than in accordance with the curriculum (Vinnervik, 2020). At the same time, it is important to have IT support staff (Ertmer & Ottenbreit-Leftwich, 2010; Vinnervik, 2020). According to Vinnervik (2020), teachers are enthusiastic about exploring programming and are committed to creating positive learning environments.

Theoretical framework

Shulman (1987, 2013) argued that when a teacher combines their pedagogical and content knowledge, a new field of knowledge, pedagogical content knowledge (PCK), is created (i.e. a teacher's unique professional knowledge). This theoretical framework can be used to describe the teacher's knowledge of how a particular content, problem or representation should be presented to different levels of learners to be understood. PCK also includes the teacher's knowledge of typical problems that pupils encounter in the subject. That is, PCK is subject-specific and pupil-level-specific and grows over the years of teaching. Over the years, PCK has been applied in numerous educational fields, for instance, science education research (Loughran et al., 2012) and *technology*

education research (Rohaan et al., 2011). The meaning of PCK and research based on PCK have been problematized (Friedrichsen et al., 2011), and the digital evolution of the past decades has spurred a direction towards technological pedagogical content knowledge (Koehler et al., 2013). However, since the Swedish context firmly places programming as an integrated part of the *technology* syllabus, we continue with PCK as part of design and *technology* education.

Since design and *technology* education has a unique tradition, (Doyle et al., 2019) developed a methodological framework for understanding PCK in enacted teaching practice. The framework presented in Fig. 1 is based on the development of PCK within science education research. On the one hand, Doyle et al. (2019) adapted Friedrichsen et al. (2011) framework in relation to design and *technology* as a subject and its teaching and learning. On the other hand, they adapted a model concerning teachers' professional knowledge (eg. Gess-Newsome, 2015). Both are further elaborated on below with respect to Doyle et al.'s (2019) descriptions.

The first part of the framework, Beliefs in Fig. 1, concerns all teachers having beliefs regarding the subject itself, how to teach it and how learning may be facilitated. It is one of the cornerstones of Doyle et al.'s (2019) framework. The field of design and *technology* education has developed over the past 30 years, and today, it focuses on technological capability and literacy. However, Doyle et al. (2019) concluded that much is still debated, and a common concern is defining design and *technology* too narrowly. Such a definition might lead to a loss of uniqueness in design and *technology* and a non-desirable limitation of the subject. Another challenge is the silent nature of *technology*. Compared to science with its revisable theories and truths, *technology* focuses on actions and applicability, relevant knowledge and problem solving, values and how to create (Doyle et al., 2019). In addition, *technology* teachers' autonomy in relation to the dynamic nature of the subject leads to very different learning goals and classroom practices. Hence, during the process of understanding *technology* teaching, it is possible to apply PCK frameworks, and general results are challenged in the specific teaching

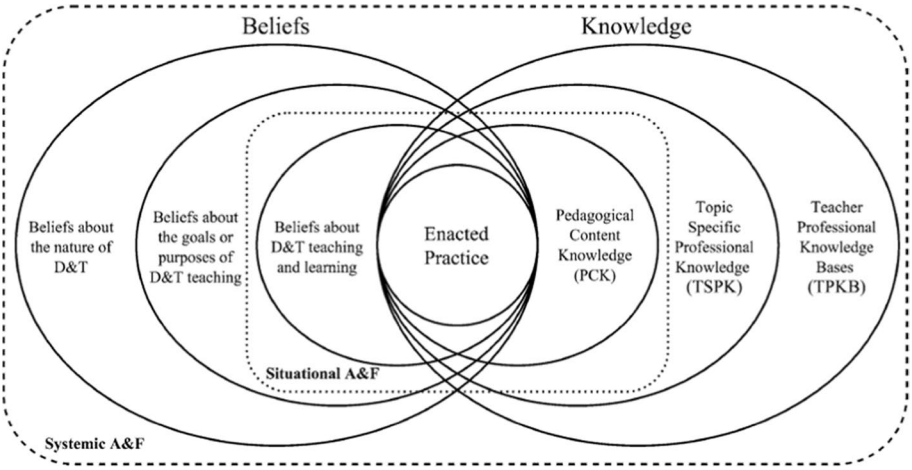

Fig. 1 Pedagogical content knowledge framework, A&F = amplifiers and filters; D&T = design and technology. Adapted from "Reconceptualising PCK research in D&T education: Proposing a methodological framework to investigate enacted practice." by Doyle et al. *International Journal of Technology and Design Education*, 2019, 29(3), 473–491. Reproduced with permission

situation: 'In D&T education therefore, understanding teachers' implicitly held beliefs about the nature of D&T and the nature of activity in D&T is of interest in understanding enacted practices' (Doyle et al., p. 477).

In the second part of the framework (Knowledge in Fig. 1), Doyle et al. (2019) refer to developments in PCK research. This part focuses on teacher professional knowledge bases (TPKB), topic-specific professional knowledge (TSPK) and PCK. TPKB include general teacher knowledge, for instance assessment and curricula, while TSPK makes disciplinary expert knowledge explicit, for instance certain strategies, representations, pupil understanding of the topic at hand and habits within the subject (e.g. Content Representation Saeli et al., 2010). Both TPKB and TSPK are shared knowledge; they are the knowledge of the profession (Doyle et al., 2019). PCK, on the other hand, refers to the teacher's experience—a kind of situated knowledge.

Continuing with Fig. 1, Doyle et al. (2019) summarize two levels of amplifiers and filters: situational amplifiers and filters of practice (situational A & F) and systemic amplifiers and filters of practice (systemic A & F). Situational A & F work both holistically (e.g. the school culture) and day-to-day (e.g. the available resources). Situational A & F are rooted in research proposing that situational forces (e.g. the teaching material, assignments and pupils) may impact pupil learning more than the individual teacher. Hence, to understand the enacted practice, situational A & F need to be addressed. Concerning systemic A & F, one important factor is the culture. If a school practises a certain pedagogy, it influences the teachers' own ideas and choices. Over time, such factors affect teachers' knowledge and beliefs. Systemic A & F is a part of teacher enculturation. As with situational A & F, systemic A & F need to be identified to understand enacted practice. In this research study, we neither followed the teachers in their authentic settings nor present results regarding the enacted practice. Still, the framework offers opportunities and theoretical concepts that help us conceptualize different choices made by the teachers as they began teaching programming as part of the Swedish grades 4–6 *technology* curriculum.

Method

Data generation and sampling

This is a qualitative study based on semi-structured interviews conducted in autumn 2018 and spring 2019, when programming was new to the Swedish curriculum. The sample was purposive (Bryman, 2014; Robson & McCartan, 2016); in this case, all the participating teachers had already taught programming in primary school (grades 4–6, pupils aged 10–12). A representative sample would not have yielded much data, since most Swedish teachers needed professional education in programming at that stage (Statens Skolverk, 2019).

In total, there were 14 participants in this study from all over Sweden. Of these, 13 teachers were contacted through social media, and one was reached through contact with several principals in a medium-sized municipality in Sweden. Of the 14 participants, two were full-time teacher trainers working in two Swedish municipalities at KomTek, whose mission is to provide coursework and increased skills in *technology* for schoolchildren and youth. This means that all the participants were working or had recently worked as *technology* teachers in grades 4–6. The 14 participants were given fictitious names following the alphabet from A–N, with male nicknames for the male participants and the same for the

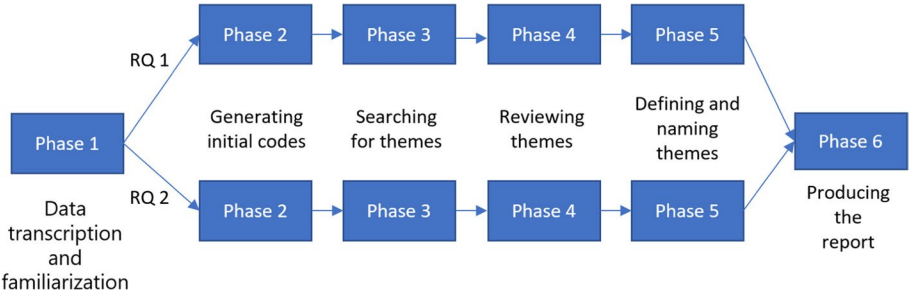

Fig. 2 Analytical process

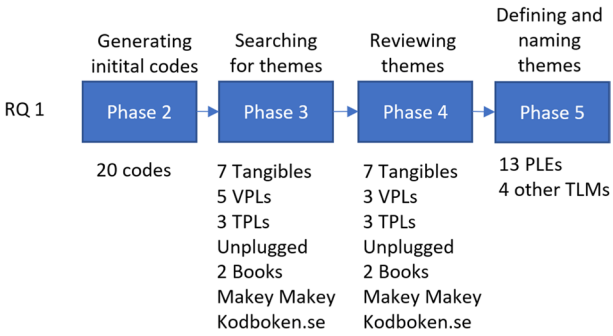

Fig. 3 Analytical process for the first research question. PLE = programming learning environment; TLM = teaching and learning material; TPL = textual programming language; VPL = visual programming language

female participants. Emma and Maria are full-time teacher educators, while Grace, Kate and Liam work part-time training colleagues. Almost all the participants have teaching qualifications, except Frida, who was soon to have one, and Jack, who has a qualification as a games programmer. Other participants with some form of higher education in programming were Anna, Bella and Camilla. Of these, Anna has 90 European Credit Transfer System (ECTS), while the other two have 7.5 ECTS. Four participants had learned to program on their own, while the rest had taken some short course(s). The participants had been working as *technology* teachers for between 4 and 29 years (Bjursten et al., 2023).

Data analysis

Braun and Clarke's (2006) analytical method was used to thematize the content (Fig. 2). In Fig. 2, the whole analytical process is illustrated. The 14 interviews were conducted, recorded and transcribed verbatim in NVivo 12 Pro by the first author. The data formed approximately 10 h of digital audio with roughly 43,000 words. Afterwards, the analysis was split into two parts to answer the first research question on the PLEs the participants used (Fig. 3) and the second research question on which factors influenced their choices (Fig. 4). In the last phase, the report was produced by combining the two different analyses.

In the second phase (the first research question, Fig. 3), we scanned the materials and identified 20 TLM items related to programming and *technology*. These included unplugged programming, Scratch, Micro:bit, GameMaker Language (GML), books, Code.org, and Kodboken.se. We then generated initial codes based on these TLM items. In the third phase, searching for themes, the 20 codes were thematized according to Humble's

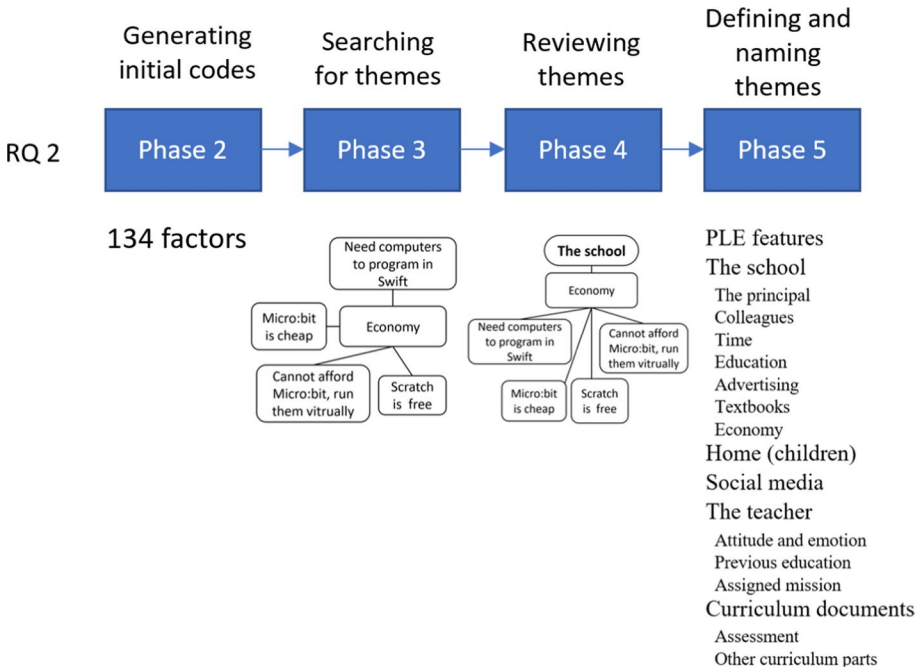

Fig. 4 Analytical process for the second research question. PLE = programming learning environment

(2021) system. This resulted in seven tangibles, five VPLs, three TPLs and unplugged programming. However, as TLM is a broader concept than Humble's programming tools, some codes did not fit the system. Kodboken.se is a website that provides programming tasks divided by programming experience, including some of the above-mentioned programming environments and both VPLs and TPLs. Also, two books were used and Makey Makey, a kit that can be both programmed and used as an input device. In the fourth phase, the three examples from Code.org, Hour of Code and Kodable were considered as underlying themes that support the parent theme of Code.org. This means that while the three examples were related to the broader topic of Code.org, they were not the main focus of the discussion or analysis. Rather, they provided additional context or details that helped support and expand upon the parent theme. In the fifth phase of the analysis, we conducted a comprehensive check to identify the TLM that met the criteria for inclusion as a digital learning environment for programming (i.e. a PLE). TLM items that did not meet the criteria were categorized as another theme. We also verified the names of the TLM items during this phase.

Analytical process for the second research question

In the second phase of the analysis (the second research question, Fig. 4), the material was searched again to look for factors that influenced the participants' choice of PLE. A total of 134 text excerpts were found. In the third phase, the text excerpts were grouped into themes, resulting in 15 themes. These 15 themes were revised in phase four, and coherent patterns were sought. We chose to bring together school factors. Furthermore,

we decided to merge statements that could be directly attributed to the teacher's attitude and emotion, previous education and assigned mission separately because these are likely to be related to the teacher's personal engagement and not constraints that exist within the school. In addition, we also chose to place statements about assessment, curriculum, syllabus and other steering documents in a different theme outside the school. These are not decided at the school itself, but at the national level. In the fourth phase, we decided to add a category for statements that addressed a challenge but were not related to any specific PLE. These statements were more general in nature (see examples below):

The principal: 'We have had very good support from our principal. She has insisted that we should work with programming, but also with the *technology* subject. It has been very important for her to see in our seasonal planning that we have planned in the *technology* subject so that the programming does not fall between the cracks. For her, it has been important'.

Economy: 'No, there are not enough resources! I have 3 computers in the classroom and 25 pupils, so if I want to do something, I have to borrow from all the other classes. And I don't have an iPad. If there's anything you wish to purchase, it must be covered by the teaching materials budget'.

Attitude and emotion: 'I am passionate about programming'.

This phase resulted in the Table 1. In addition, the table presents the number of participants who mentioned each theme rather than how many times a single participant mentioned a theme.

Table 1 Factors influencing teachers' choice of programming learning environment (PLE)

Factors	Total
PLE features	11
The school	
The principal	4
Colleagues	7
Time	1
Education	2
Advertising	1
Textbooks	1
Economy	8
Home (children)	1
Social media	6
The teacher	
Attitude and emotion	7
Previous education	12
Assigned mission	4
Curriculum documents	
Assessment	12
Other curriculum parts	9

Table 2 Teaching and learning materials

PLE[a]/Other TLM[b]	Category (Humble, 2021)	TLM	Mentioned by participant	Total
PLE	VPL[c]	Scratch	A B C D E F G H I K M J	12
PLE	VPL	Code.org	A B C H I K L M N	9
PLE	Tangible	Micro:bit	B E F G K M N	7
PLE	Unplugged	Unplugged	B C E G K M N	7
PLE	Tangible	LEGO	D G H L M N	6
PLE	Tangible	Blue-Bot®	G K L M N J	6
PLE	Tangible	Sphero	A K	2
Other TLM	n/a[d]	Books	A C	2
Other TLM	n/a	Makey Makey	B I	2
Other TLM	n/a	Kodboken.se	A G	2
PLE	Tangible	M-bots	J	1
PLE	Tangible	Ozobot	K	1
PLE	Tangible	Quirk Bot	M	1
PLE	VPL	ScratchJr	E	1
PLE	TPL[e]	CodeMonkey	D	1
PLE	TPL	GameMaker Language	J	1
PLE	TPL	Swift	L	1

[a]*PLE* Programming learning environment

[b]*TLM* Teaching and learning material

[c]*VPL* Visual programming language

[d]*n/a* Not applicable

[e]*TPL* Textual programming language

Results

The results describe which TLM primary school teachers use in teaching programming in *technology* and the factors influencing their choices. The results are organized into two sections. First, we present the results of the first research question about the TLM used (Table 2). Second, we present Table 3 on the aspects that influenced the teachers' choice of PLE. All the interviews were conducted in Swedish. All the excerpts were translated to English by the first author. Since the teachers' fictitious names are created according to the alphabet, it is possible to infer which teacher is shown in the tables.

Included teaching and learning materials

Table 2 illustrates the TLM used by the various teachers in their classrooms. By referring to the alphabetical names assigned to the teachers, it is possible to determine which TLM each teacher uses. This also highlights the diversity of TLM utilized in *technology* education across primary schools in Sweden. The table has been arranged in descending order according to the frequency of TLM usage among the participants.

The participants use many different TLM items. Looking at how many participants mention each TLM item, we see that 12 participants mention the block programming language

Table 3 Factors influencing teachers' choices of teaching and learning material

	General statement	Tangible					VPL			TPL			Unplugged programming	Kodboken.se	Total
		Blue-Bot®	LEGO	M-bots	Sphero	Micro:bit	Code.org	Scratch	Scratch Jr	Code Monkey	GML	Swift			
PLE features	N	JENLF	MF	J	F	CM	ABCKL	J			J		CGKN	A	11
The school															
The principal	FGDC														4
Collegues	N	N	DN			N	CN	AE					CGKN		7
Time	L														1
Education		N			A	N									2
Advertising										D					1
Textbooks	B														1
Economy	JLKHNB	L	F		F	LF	K	K		D					8
Home (children)								D							1
Social media	AKLBE					B	I	AB EI							6
The teacher															
Attitude and emotion	JIN					F	K	FA			J		M		7
Previous education*	JBICL	M	D			GFCI	NMA	HKC							13
Assigned mission	MGLN					N									4
Curriculum documents															
Assessment	KML-NJHEDGC		L			NB	M	IEB	E						12
Other parts of curriculum	KICJHM	LNG	L		K	K	L	K				L			9
Total	14	7	5	1	3	9	8	10	1	1	1	1	5	1	

*Additional programming languages: Anna, C++and JavaScript; Jack, Aurdino and Python

GML GameMaker Language, *PLE* Programming learning environment, *TLM* Teaching and learning material, *TPL* Textual programming language, *VPL* Visual programming language

Scratch; almost as popular is Code.org, mentioned by 9 participants. However, Jack said that block programming does not reflect real programming and should therefore not be used. This means, in fact, that Liam, Nora and Jack do not use Scratch in the classroom. However, Liam uses another block programming language by having his pupils program in Code.org and the tangible Blue-Bot® with an application on the iPad. Nora's pupils, on the other hand, get to use block programming in Micro:bit and Code.org.

Half the participants use Micro:bit. As a rule, it is used together with a physical microcontroller, but one teacher only uses it virtually. Micro:bit is used by participants in very different ways. Most let their pupils program the microcontroller with the attached block programming language, but three participants said that textual programming occurs, mainly with JavaScript. Half the participants use unplugged programming, often in combination with an introduction to programming. Six participants mentioned that the pupils program LEGO robots, which makes it slightly more prevalent than the Blue-Bot®, which five participants use. Besides JavaScript, which was mentioned in relation to working with Micro:bit, three different TPLs were mentioned, two of which are open programming environments, GML and Swift, while CodeMonkey is closed, that is, with ready-made tasks to be solved.

Although only a small number of participants used Makey Makey and ScratchJr, these tools are still noteworthy. One participant who mentioned Makey Makey noted that it mainly involves wiring, but programming functions are also available. However, there was no evidence to suggest that the participants actually used the programming feature of Makey Makey. ScratchJr, on the other hand, is a simplified version of Scratch and was mentioned only in relation to assessment.

When the participants described the TLM they use in teaching *technology*, we did not always find an explanation of why they use it, meaning that some TLM items that appear in the results here cannot be found in the results of research question two.

Factors influencing teachers' choices

Table 3 shows which teachers talked about the factors that influenced them to use a particular PLE. The table does not describe how many times a teacher mentioned the same factor for the same PLE. Rather, the table should be seen as a qualitative evaluation of the diversity of PLEs used in the programming classroom.

As an example, Nora chose the Blue-Bot® because of her colleagues. Liam also chose the Blue-Bot® but the school's economic situation affected his choice of TLM. By comparison, the economy affected teacher Dora's choice to use CodeMonkey, since the school had a good economy in those days.

PLE features

The participants told us that several different PLEs are useful for starting programming. Emma found Blue-Bot® to be a clear and concrete environment. She said that Blue-Bot® is not something that years 4–6 should use continuously, but as an introduction, it is excellent, especially if programming is new. Others made it clear that unplugged programming is good for an introduction because it is simple and practical, while some suggested that pupils should start with Hour of Code. Blue-Bot® can also be used by programming the little robot via an app, and both Liam and Nora stated that this can also be included as an introduction to programming. On the one hand, the participants stated that these

PLEs provide an understanding of the programming conditions, that accuracy and clarity are important and that PLEs increase understanding. On the other hand, the participants seemed to disagree on what is easy and good for pupils. Jack described how his pupils use GML because it is easy to code and explained that GML is designed to make games very quickly. Furthermore, Jack expressed the reservation that Bee-Bot® and Blue-Bot® should not be used in middle school, as they are too simple. He said that his pupils used the more advanced M-bot robot once before he switched all his teaching to GML. Following this, Jack also made it clear that he does not think that Scratch reflects real programming. Hence, Jack neither uses Scratch, Bee-Bot® nor Blue-Bot® although he mentions them.

The participants said that a PLE can be both flexible and inflexible. Camilla explained that the choice of the hardware Micro:bit was due to its flexibility, as pupils can program it with both block and text programming. Maria stated that this was the most important reason why they chose Micro:bit in the first place. In contrast, Camilla stated that Code.org (the website where Hour of Code is located) is inflexible and pupils lose creativity because the exercises are static. Choosing a programming language can also provide the flexibility of choosing hardware, which Maria explained in relation to when they decided to teach the JavaScript programming language:

> We have found that this [JavaScript] is what the pupils know, so it has been easy. If we say Python or C, they [the pupils] don't know it. It's an industrial language, whereas JavaScript is familiar to most people. They see it on websites, and they say they've seen it before. Then, of course, it also depends on the different hardware that we have here. LEGO, Micro:bit, and robots are also supported by JavaScript. It's been pure simplicity because Java works on many devices.

Furthermore, to enable pupils to progress in programming and to see how a programmed robot moves, Nora said she would like the school to buy new hardware that is programmable. Her pupils control the virtual Blue-Bot® through iPad applications and therefore have not seen how a robot moves.

The school

The school includes seven sub-themes: the principal, colleagues, time, education, advertising, textbooks and economy. While the seven sub-themes under the school are relatively clear, one sub-theme may require further clarification or explanation. Specifically regarding education, we gathered text excerpts from the participants about the educational activities *initiated* by the school. Each sub-theme includes text excerpts that provide further insight into and context for how the school shapes programming education in *technology*.

Support from the principal is closely linked to the assigned mission theme because some participants had been tasked with establishing digital support or digital programming in their school. Support from the principal, however, was not very visible in our data, as only two participants mentioned it. Nora said that her principal supports the implementation of programming but does not interfere with which PLE they choose. Moreover, Frida pointed out that the principal not only provides support but stresses the importance of programming and *technology* being included in the year plans. In contrast, others perceived no support from the principal or, as Camilla put it, 'It is up to each individual to acquire the skills'.

Colleagues provided good support to the participants in choosing a PLE. IT educators, science and technology inspirators and first teachers (i.e. a teacher with particular skills in

this area) provide advice and train their colleagues. Nora receives in-service training when there are workplace meetings and study days and said, 'We learn a little bit at a time'. She said that all the PLEs she uses in teaching had been reviewed, and everyone in the school had also received training from the company they bought the hardware from.

Furthermore, other colleagues (i.e. regular teachers) also helped and supported each other in finding a PLE suitable to use in teaching. However, not all the teachers have a colleague or other support in their schools. Irene teaches programming in all classes at her school, and she argued, 'And of course, it would have been good if I had a parallel teacher to compare notes with'. For instance, not all municipalities have a KomTek department that supports teachers' *technology* teaching.

Liam was the only participant to talk about how time can dictate the choice of PLE. He explained that it is important that any teacher is given time to make the change and not to rush through anything. Liam was given the task of investigating how programming could be carried out at his school by his principal about a year and a half before programming was introduced into the curriculum.

> We bought different robots, and then I sat down and evaluated which ones we should continue with, which languages, and which ways we should program. /---/ First, I had to think and test with my class with programming. /---/ After that, I had workshops where I showed the teachers. /---/Then my task was to think about what to do in the different classes to make it happen.

Liam explained that it is important that teachers and pupils both feel comfortable and that it is not too time-consuming to get into programming for it to work. It cannot be too difficult for teachers. The principal needs to guide and provide the conditions. Liam believed that the school does not need to buy the latest programming equipment: 'You can do cool stuff even if you have some older programming equipment', he added.

Only one participant mentioned that they had received training through the school they work at. As reported earlier, Nora has both good support from the principal and a colleague who has a mission to implement programming in *technology* education. At her school, all teachers receive in-service training. However, the company they bought the hardware from had been running the courses. Anna described another form of in-service training. As she initiated contact with KomTek in the municipality where she works, she also received in-service teacher training regarding the Sphero Robot, as her class worked with it.

Only one teacher talked about how advertising influenced her choice of PLE. Dora reported that her school purchased an annual subscription to CodeMonkey. Nevertheless, her school could not afford to buy that PLE twice.

Since programming in Swedish schools is new, it is not surprising that textbooks have not been updated with the new content. Bella described this difficulty. On the other hand, she did not want to be guided by a textbook. She continued by explaining that she does not see this as a problem. This suggests that she does not want to be guided on how teaching can be conducted.

Some participants' schools were short of money, while others seemed to have plenty. The teachers working in schools where there was a shortage of money said that it often dictates which PLE they use in teaching. Those teachers use Code.org and Scratch because it is free and Blue-Bot® and Micro:bit because they are cheap. Frida confirmed that Micro:bit can be simulated on computers; there is no need to buy it. On the other hand, Frida argued that she wanted to buy Sphero: 'Working with robots that have to complete some kind of track/…/That's the goal I want to reach.' Likewise, Bella emphasized her situation as quite desperate. However, Henric made it clear that there was enough in relation to the

limited timetable that the *technology* subject has. Other teachers agreed. For example, Nora explained:

> We have one computer for two pupils, and they are the kind of computers that other pupils have used before, so it takes a very long time to start. But once the pupils are in and the internet is working, so I think the resources are okay here.

In contrast, the teachers working in schools where there is no shortage of money said that more hardware needed to be purchased. Liam said that 20 computers are needed and that is a matter for the management, but he explained that if he can justify it, this is put into the budget and then purchased. At the school where Jack works, this seems to be already underway, as the school planned to purchase desktop computers to allow pupils to program in 3D. A more modest investment was described by Kate. She explained how her school had received funding for a digitization project through a university. This enabled them to purchase hardware. They then created boxes similar to the NTA (i.e. Science and Technology for All) concept for programming. Kate added that this increased the equality of teaching, as all pupils have the opportunity to try programming. Similarly, Bella confirmed that she also has some support from a nearby university, where they have started a project that aims to increase interest in science and *technology*. The project is funded by a number of independent organizations.

Social media

Most of the teachers reported that the internet, YouTube, Facebook and other social media are important for spreading the word about which PLEs are good to use. Anna described how she heard about Scratch.

> I'm a member of a lot of Facebook groups, and most of them are Finnish groups. One is about how to use the iPad in teaching. One group is about programming and then there's a very big group that's about everything in primary and secondary school. In Finland, programming came into the curriculum earlier and then there were a lot of teachers crying out for help – 'Now it's bad. I have no idea what I'm doing' and then there were many colleagues who said, 'Start with this, this is really easy'. And when I read those discussions, I thought, I want to try this too.

As several participants confirmed that they are alone in the school in programming with pupils, they turned to social media as a channel to get quick help on programming or if they felt they needed to discuss something with a programming-savvy colleague. Scratch was the PLE that most participants described learning about from social media.

The teacher

The teacher is a key figure in the educational process, and three important sub-themes relating to teachers are attitude and emotion, assigned mission and previous education. Attitude and emotion refer to the teacher's demeanour and approach to the new content on programming, while assigned mission relates to the mission or objectives given to the teacher by the principal. Previous education refers to the teacher's academic and professional background, including any education they received before computer programming was introduced into the curriculum or any education that the principal or the school may not have provided or supported.

In some cases, the teacher's attitude and emotions determined the choice of PLE. Scratch and Micro:bit, according to the participants, are easy to use. In addition, Maria said that the teachers who come to KomTek are comfortable with unplugged programming (i.e. programming without digital devices). The Code.org website was also mentioned as simple and accessible.

Most of the participants told us that it was their own interest that seemed to be the main driving force for them to start programming. They mentioned not only Blue-Bot® and Hour of Code, but also that anything to do with *technology* is interesting. Two participants, Bella and Jack, stressed that they are passionate about programming. Anna, who said that she started Scratch in her class because she had heard about it, mentioned that she is interested but clarified that she 'tested' when she programmed with her class for the first time.

However, it is sometimes easier said than done. One teacher argued that it is a problem being a primary school teacher because they have to teach so many subjects:

> And if I had only been familiar with the methodology and the *technology*, I would have found it enjoyable. But I feel that in [grades] 4–6, we have so many subjects that we have to be good at, so there is not enough time, so we can't have any special interest. We have to keep ahead and be up-to-date in all subjects, and we can't be super good at, for example, *technology* or *history*.

Seven participants had some kind of assigned mission to help teachers at their school or the whole municipality teach programming. Here, we found all sorts of preparation, such as Liam's list of programming tasks, corresponding to each school grade, along with Maria and Emma, who work as full-time teacher trainers at KomTek. Mostly, the participants talked generally about their mission. For instance, Grace said that this is partly because she is interested, but also because she 'has a mission to develop such things'. Nevertheless, Nora was very specific about her role in leading the school forward in this new content: 'I led the digitization project at my school'. In contrast, Jack has a special assigned mission, as he was hired solely to teach *technology*, especially programming.

The influence on which PLE the participant uses in their teaching was not always related to previous experience. Although Anna has 90 credits in programming and knows both JavaScript and C++, she had programmed in Scratch with her pupils. Moreover, her previous training made her confident in programming.

> I have so much basic knowledge, the colleague had no idea about programming, and she thought it was really hard. It's probably hard for regular classroom teachers.

For one participant, programming was not an obstacle but rather an opportunity. Jack has a professional background as a games programmer and teaches textual code in GML to pupils from grade 3. He admitted, 'It can be a bit tricky, but it works just fine'. He noted that he had gained a lot of knowledge about what works and what does not work in the classroom. GML is the TPL his pupils use. Jack believes that GML, which is similar to JavaScript and C#, paves the way for his pupils' further education as that programming language provides a good grounding.

Correspondingly, Kate had attended a short course in creative and digital storytelling more than five years ago, which inspired her to use Scratch in the classroom. Most participants had started to implement programming in their classrooms before it was mandatory. However, not all had prepared their whole school as Liam did:

> When I heard that programming was going to be inserted into the curriculum, we [the school] were a year to a year and a half ahead. We had already bought different

> robots, and I evaluated which ones we should continue with, which languages, and in which ways we should program.

Several described how the courses they took influenced their choice of PLE, which Camilla made clear influenced her teaching.

> The course was good because you got some tools; it was concrete. We got to practise the tools that you could use with the pupils. However, 7.5 credits are not that much, so you realize how little you know when you have completed a course like that. But it was a good foundation to be able to work with programming with the pupils in the classroom and in the meantime, you learn more and more. /---/ I started with them [these particular PLEs] entirely because I took this course. The literature and assignments in the course were directed to these software. And I was expected to do a project with my pupils.

Curriculum documents

Assessment and other parts of the curriculum are the two sub-themes that fall under the larger theme of the curriculum. Assessment refers to the text excerpts that in some way relate to knowledge requirements, while other parts of the curriculum encompass all the different areas of study beyond knowledge requirements, such as core subjects, electives and extracurricular activities. Together, these sub-themes offer a comprehensive look at how the curriculum impacts programming education in *technology*. Given that assessment is a fundamental part of the educational process, it is not surprising that the participants discussed it in their responses. Additionally, as part of the enquiry, assessment was one of the key areas of interest, and the participants' insights offered valuable perspectives on this critical aspect of education.

When the participants were asked about assessment of programming in *technology*, they did not describe it in relation to any PLE, and therefore opportunities for assessment do not control the choice of PLE. During the interview, Bella explained how an assignment could be designed to allow pupils to choose one of the PLEs they have used in teaching. This was also described by Nora; however, she gave a broad and vague example of Micro:bit: 'If you have worked a lot with Micro:bit, then you can give them an assignment'. Another suggestion came from Emma, who said that the teacher could show the pupils what happens when a program is run and then let the pupils program it. Emma went on to say that programming can be used to demonstrate other knowledge: 'For example, the water cycle. Can you illustrate that in Scratch? Yes, you can', she continued.

The absence of knowledge requirements was something that Camilla thought about. In *mathematics*, she assessed tasks in which pupils were given the choice of presenting a task using either analogue or digital aids. However, whether the pupil chose to present an analogue or digital result was not assessed, only the mathematical content. In the *technology* subject, she was more uncertain:

> Yes, it's a bit weird that it's not in the knowledge requirements. On the one hand, you are supposed to teach programming, but on the other hand, you don't know what they should be able to do at different grade levels. How long a program should you be able to write or what should you program to reach a certain grade level? There's no such thing. It's more that you have to know that you can do it. /---/ It's all about understanding how computers...understanding the accuracy...

> the computer just does what I say. So, if it [the computer] gets it wrong, I'm the one who got it wrong. But then exactly in any assessment...I don't know...

Grading and assessment are done on the whole or on other elements that the participants felt the programming could demonstrate, which some described as problem solving, or as Liam described it, 'Those who have the energy to sit and try and try'. Most of the participants were still unsure whether it should be assessed. Nevertheless, if programming should be assessed, they were hesitant about how assessment should be done, what should be assessed and what can be classified as good or less good. Emma was confident that this would be clearer in the next curriculum because she thought that there would be knowledge requirements then.

Despite the lack of clarity in the curriculum, it seems that the majority of the participants thought that pupils in grades 4–6 should use block programming (VPL). Kate said that it is 'natural' to do block programming in grades 1–6. She said that this is good because pupils do not need to be able to type on the keyboard, but can move pre-programmed graphical objects. Both Jack and Henric said that pupils should learn several programming languages. Although it looks like Jack and Henric thought alike, there was a big difference, as Jack only uses TPL, while Henric only uses VPL. Henric argued:

> ...it says in the curriculum that they should know some different programming languages and how to control something using programming...

Some participants also mentioned that they think it is bad that every school has to develop its own concept for programming (e.g. which PLE is suitable for which grade). Kate explained, 'You start with the robots [Blue-Bot® or Bee-bot®] and you press them, and you see immediately what happens'. Because the curriculum is unclear, the participants felt that they were freedom to choose the PLE they wanted.

Some of the participants also referred to other parts of the curriculum. Jack argued that the product that pupils create with programming is a form of entrepreneurial learning, which can be found in the curriculum under the fundamental values and tasks of the school and in the *civics* syllabus (Statens Skolverk, 2018). In contrast, Liam argued that he uses Blue-Bot® together with iPad and Code.org for pupils to understand algorithms and follow instructions. Algorithms are part of the core content of mathematics, and instructions are found in the Swedish-language curriculum. However, he also confirmed:

> With the LEGO Mindstorm PLE, I wanted to cover the knowledge requirement to create something that could move. That was one of the thoughts; plus, it would be a bit challenging [for the pupils].

This confirms that there were also thoughts on the curriculum in the *technology* syllabus (see the introduction). Henric also mentioned the programming content of the *technology* syllabus, '… it says in the curriculum that they should know some different programming languages and how to control something using programming'. However, in the *technology* syllabus 'some different programming languages' refers to grades 7–9.

Jack stressed the importance of multiple programming languages as an important part of programming. He thought that pupils should learn several programming languages. Either way, Jack probably meant several TPL, while it is unclear whether Henric meant VPL or TPL in his statement.

Discussion

The discussion is divided into two parts. First, we answer the question of which PLE the participants use. This is followed by a discussion of the challenges that the participating teachers have overcome.

What teaching and learning material do Swedish teachers in grades 4–6 include in technology?

The participants provide pupils with a variety of opportunities for extended learning in programming by offering instruction with multiple PLEs. Half the participants use Micro:bit in their teaching and all but Jack teach some VPL. Scratch is the most common VPL, as Humble (2021) also shows. An unexpected result, which differs from Humble, is that one participant, Jack, only uses TPL in his teaching. In addition to Jack, both Dora and Liam have used TPL, but for Dora, finances put a stop to further programming with CodeMonkey. A possible explanation for why Jack uses TPL is his previous training as a games programmer.

Although there are several references to Blue-Bot® in Humble's (2021) material, the results of this study show that Blue-Bot® is not very relevant to grades 4–6. Blue-Bot® is used as an introduction to programming before the teaching moves on to more advanced programming. Jack strongly opposed using Blue-Bot® in the classroom at all. Following this, Fessard et al. (2019) reported uncertainty regarding the extent to which physical programmable objects contribute to increased learning outcomes. In contrast, while some studies have suggested that these tools can enhance pupils' learning (Przybylla & Romeike, 2015), the evidence is not yet conclusive, and further research is needed to determine the effectiveness of these tools in promoting learning in different contexts.

Many participants in this study use some form of internet resource; however, one participant felt that the closed software Code.org does not allow for spontaneous programming and allows only inflexible teaching opportunities, as all closed programming environments are static in form (Rich et al., 2022). Nonetheless, the popularity of these programming tools may be attributed to the fact that they are free to use. In Humble's (2021) data, Code.org is classified under the VPL category. However, it should be noted that the web page also offers opportunities for pupils to program in TPL, making its placement in Humble's system inexact. Thus, although our data do not provide evidence that Code.org has been used by participants for their pupils to program in TPL, the possibility exists and warrants discussion. In contrast, Rich et al. (2022) identify Code.org as software and as a software/game development kit in their appendix, which we can agree upon. Moreover, Rich et al. discuss the advantages and disadvantages of VPL and TPL, yet it is not used in the categorization. For this reason, by using both VPL and TPL in the categories, we can better understand the capabilities and limitations of different programming tools and make informed decisions about their use in education. Overall, it is important to consider all aspects of programming tools to make informed decisions about their use in education. Furthermore, this is also evident with Micro:bit, where pupils have the opportunity to engage in a TPL, which was evident in our data. As *technology* teachers are autonomous and the subject is dynamic, it results in very different classroom practices and learning outcomes (Doyle et al., 2019). When considering programming and TLM as part of the Swedish *technology* curriculum, it is obvious that teachers' different beliefs and choices

result in different teaching opportunities and learning outcomes for our pupils. However, what teachers do in their classrooms has not been studied here. Still, the choice of PLE is an example of situational A & F, implying that both the reform itself and the teachers' choices of PLE impact what the pupil may learn in programming within *technology*. Apparently, most Swedish grade 4–6 *technology* teachers include unplugged programming, tangible objects and VPL.

What factors influenced their choice of PLE?

In the data, we saw many references to participants using a particular PLE for its properties. Since we asked in what way they use a PLE, the interviews revolved around how they use the PLE, which may be what Vinnervik (2020) describes as 'tool-centric'. Surprisingly, the participants did not discuss any programming concepts when talking about interacting with a PLE. Admittedly, this result could possibly be explained by how vaguely programming is described in the curriculum (Statens Skolverk, 2017a, 2018). Moreover, it is crucial for the participants to engage in discussions related to programming concepts, as they have been shown to be essential in programming education (Saeli et al., 2010). Therefore, if the participants do not engage in such discussions, it may hinder their ability to fully grasp the fundamentals of programming and hinder their progress in this field.

Few participants mentioned that the principal supported the introduction of the new core content. Camilla's statement that it was up to each individual to acquire knowledge of programming is an example of that attitude. This is in line with the findings from Vinnervik (2020), where support from the principal was unclear. School finances are the responsibility of the principal, and major investments are requested, while others have solved financial shortcomings by using what is free of charge. In contrast to the unclear support of the principal, colleagues seemed to provide support. In addition, it was not only colleagues from their own school who mattered, but also support via other colleagues from social media. This underscores the value of social media as a tool for supporting teacher professional development and the potential benefits of incorporating it into teacher education programmes. It also connects the study's findings to the broader literature, which holds that opportunities to discuss in different forums are relevant (Ertmer & Ottenbreit-Leftwich, 2010; Finger & Houguet, 2009; Yadav et al., 2016). It is obvious that the teachers' effort to work within the new content relied on different situational A & Fs (Doyle et al., 2019). However, with respect to Doyle et al.'s summary of systemic A & F, we found little or no evidence of such in the results. With respect to this, the situation in Sweden and the sample itself may be one possible explanation for this fact. The sample included interested teachers who had begun teaching programming within the subject of *technology* prior to it being mandatory. No culture existed; instead, these teachers may have been in the process of developing different school cultures. In addition, there is only one requirement in the *technology* syllabus: 'Controlling pupils' own constructions or other objects by means of programming' (Statens Skolverk, 2017b). Hence, teachers are autonomous in choosing both the PLE and the programming content to include. Still, with respect to Doyle et al.'s (2019) descriptions of systemic A & F, we found the boundaries for the educational context and the meaning of culture somewhat problematic. For instance, are social media and advertising an integrated part of today's educational context, thus impacting pedagogical practice and the choices teachers make? What we can see is that home, advertising, textbooks and time did not seem to matter in the choice of PLE. However, this study was conducted prior to programming being mandatory. Therefore, we find it important to take social media and

advertising into consideration as part of the systemic A & F. As can be seen in Table 3, one participant mentioned that her child had inspired her to use Scratch, but no one else mentioned home. Dora, who received advertisements about CodeMonkey, was able to use it because the school could afford it at the time, but she did not see the possibility of purchasing it multiple times. Dora's situation is supported by the findings of Player-Koro et al. (2018), who point out that certain IT providers see schools as a potential market for their products. However, Dora's situation also highlights the economic challenges that many participants in similar circumstances face. The issue of limited financial resources can contribute to the problem of unequal access to educational technology. Specifically, schools with limited budgets may struggle to provide their pupils with the necessary resources to succeed. This could result in a situation where only schools with sufficient resources are able to provide their pupils with certain educational technology, leaving pupils from less affluent families at a disadvantage.

In our study, Jack exemplified a teacher who possesses both content knowledge and, digital and professional resources, thereby providing his pupils with a comprehensive and thorough technological education compared to the other teachers and their respective pupils. It is noteworthy that pupils who lack access to the same resources as Jack's pupils may not acquire the same capacity to learn and develop crucial skills within the *technology* subject, potentially contributing to a widening of the achievement gap and the exacerbation of existing educational inequalities.

During the interviews, all the participating teachers, except Emma, shared their personal perspectives. However, despite not explicitly discussing her occupation as a full-time teacher trainer, Emma's viewpoints align with the themes of attitude and emotion, previous education and assigned mission that emerged in the study. Therefore, her insights contribute to a more comprehensive understanding of teaching experiences and challenges. Although evidence in the interviews suggests that some participants had not discussed the topic, the absence of a response from one participant may be due to limited questioning or a missed opportunity. Nevertheless, this supports Vinnervik's (2020) study, in which he explains that teachers are passionate about the new content and are fostering a love of learning programming among their pupils. Admittedly, although several studies suggest that teachers need time for professional development and networking to effectively incorporate educational technology into their teaching practices (Ertmer & Ottenbreit-Leftwich, 2010; Finger & Houguet, 2009; Yadav et al., 2016), this study's results suggest that a passion for programming and a willingness to take on new challenges can also be important factors. As the subcategories discussed here demonstrate, a teacher's interest in *technology* and programming, as well as their mission and experience in the field, can all play a role in their ability to successfully integrate educational technology into their teaching. Further research is needed to explore these factors in more depth and to identify additional strategies and resources that can support teachers in effectively incorporating technology into their teaching practices. As noted above, it is obvious that the knowledge domains (Doyle et al., 2019) of programming as part of the Swedish *technology* curriculum indicate a practice in which shared knowledge (TPKB and TSPK) will develop as more teachers begin to teach programming. This, for instance, is apparent with respect to assessment (TPKB). There are ambiguities concerning assessment, and even though 12 of the 14 teachers talked about assessment, only 7 linked assessment and their choice of TLM.

Although we did not specifically analyse the interviews according to Fessakis's (2019) axes, we will discuss each axis in detail to examine how participants' responses can be aligned with each axis, based on our interpretation of the data. This is important, as these five axes can make it easier to choose suitable tools depending on the teaching

goals and can provide a framework for understanding the factors that influence educators' decisions.

First, with respect to axis A, it became apparent that the participants did not explicitly discuss the abstraction level of programming languages, although one participant, Jack, acknowledged the potential even for third-grade pupils to use TPLs. However, such a possibility may be contingent on Jack's individual background. Conversely, most participants expressed the view that block-based coding (VPL) should be introduced in grades 4–6. It is worth noting that such opinions may also be influenced by the participants' professional and content knowledge.

Second, regarding axis B, the data did not support teachers considering pupils' developmental levels when selecting instructional materials. On the one hand, this lack of consideration may result in instructional materials that are too challenging or too simplistic for certain pupils, leading to suboptimal learning outcomes. To address this issue, a more nuanced approach to material selection that takes into account pupils' developmental levels and learning needs is recommended. On the other hand, it's essential to empower teachers with the autonomy to adapt and customize materials to better suit the individualized needs and levels of their students. Teacher emancipation in the adaptation of learning materials to pupils can play a vital role in enhancing the overall learning experience.

Third, while we did not have any specific data related to axis C, our analysis of the PLEs used by participants suggests that many of the programming models exist in primary programming in *technology* in grades 4–6. For example, the participants used a variety of PLEs, including Scratch and TPL, which provide different programming models, such as event-driven and procedural programming. By using PLEs that support different programming models, the participants can tailor their instruction to meet the needs and interests of their pupils and help them develop a range of skills and competencies.

Fourth, we found that the participants' responses regarding axis D, the number of supported programming languages, did not always align with Fessakis's (2019) perspective. Fessakis (2019) suggests that having more supported programming languages is an important factor in choosing the correct PLE. However, Maria's response contradicts this viewpoint, as she stated that they chose Micro:bit precisely because it has Java, which can be used in several programming environments, rather than the number of programming languages supported by the PLE. This suggests that, while the number of supported programming languages may be an important factor for some educators, it is not necessarily the deciding factor for all.

Lastly, the abstraction level of the programming process, axis E, is an important factor that affects pupils' perception of the meaning of programming. The results demonstrate that the participants did not discuss this axis explicitly, but their responses provided some insights into how this axis can influence their choice of PLE. For instance, some participants reported using Blue-Bot® first, a simple tangible robot that allows pupils to program it and see how it moves, and then moving on to another PLE that is more abstract. This suggests that the participants recognized the importance of gradually increasing the abstraction level of the programming process to help pupils better understand programming concepts.

In summary, our discussion of Fessakis's (2019) axis has shown that educators do, in some cases, consider these five axes when selecting PLEs. The axes offer a valuable framework for aligning tools with teaching goals and identifying factors that inform decision-making. However, while considerations such as compatibility and specific features are important, it is essential to take a holistic approach to PLE selection. Further research

could explore the potential impact of these axes on educational outcomes or investigate additional factors that influence educators' decisions.

Conclusion and implications

The aim of this study was to characterize the choices of teachers in grades 4–6 (pupils aged 10–12) in relation to the selection of TLM for programming education in *technology*.

The findings highlight the significance of three key factors in choosing a learning environment: the characteristics of the PLE, the teacher's own education and the curriculum. These three components emerged as crucial determinants of effective learning experiences, while other factors had a relatively limited impact. Undoubtedly, the role of teachers in facilitating learning experiences and their own level of education cannot be overstated. Equipping teachers with the necessary knowledge and skills is essential for creating an optimal learning environment. Moreover, it is crucial to recognize the role of teachers' preservice education in preparing them for effective programming education.

The analysis underscored a clear focus on Scratch and other VPLs, which is apparent in the limited range of offerings for learners. This narrow focus on Scratch may pose certain limitations in terms of diversifying the learning opportunities available to pupils. It is worth considering how programming education in *technology* could be expanded to encompass a wider variety of tools and approaches beyond Scratch to broaden the learning opportunities available to pupils.

When our study was conducted, there was a limited range of PLEs. Now, the market has exploded and a huge number of PLEs are available—internet-based or installed on pupils' own digital devices, open or closed, with a tangible robot or not. Given the multitude of options available, choosing the right PLE can be challenging (e.g., choosing an educational robot, Evripidou et al., 2022), particularly when there is a lack of support from school principals or a specific curriculum guiding the decision-making process.

It is worth noting that neither Doyle et al. (2019) nor Gess-Newsome (2015) explicitly mentioned the role of colleagues as situational A & F, which could be the result of their primary focus on the knowledge base of PCK. However, we firmly believe that the role of colleagues as situational A & F should be considered. The collaborative interactions and support provided by colleagues can significantly impact the implementation and refinement of instructional strategies. Colleagues serve as valuable resources that can enhance teaching effectiveness. Therefore, we advocate for further exploration of and research into the role of colleagues, as the presence and impact of situational A & F in the classroom remain difficult to assess due to the limitations of our semi-structured interview approach. On the other hand, programming in *technology* currently lacks an established culture, which can be explained by the novelty of the core content. However, the participant teachers were actively working towards creating a new culture (i.e. systemic A & F), and they did not feel constrained by a heavily controlled curriculum.

Ultimately, this study emphasizes the importance of understanding the support systems available to teachers and the influence of systemic factors in implementing programming education. The findings underscore the significance of professional support, the potential benefits of making use of social media for professional development and the need to address economic disparities to ensure reasonable access to educational technology.

Furthermore, the curriculum plays a vital role in shaping teaching. Due to the broad description of programming in governing documents and the fact that the syllabus in

technology currently lacks defined abilities and knowledge requirements that could easily be used to assess programming tasks, we have begun a study to investigate how teachers teach and assess pupils' programming tasks in relation to the curriculum (Statens Skolverk, 2022).

Future studies should explore the cultural implications of programming education in *technology* (c.f. Systemic A&F, Doyle et al., 2019). With a wide range of schools and educational settings, it has become crucial to investigate the cultural aspects that have emerged within these contexts. Understanding cultural dynamics will help gain insights into how programming education is implemented and received across different schools.

It is also important to address the concepts that should be included in programming education for pupils in grades 4–6. Further research should evaluate whether the programming concepts identified in Saeli's (2010, 2012) study are suitable for pupils in these grades. It is necessary to examine the relevance and applicability of programming concepts within the Swedish context, considering the unique characteristics of the *technology* syllabus and the autonomy enjoyed by teachers. Additionally, given that some PLEs may not support all commonly used programming concepts, it becomes crucial to assess their compatibility with the learning objectives for pupils. By conducting such studies, researchers can problematize relations between the curriculum, programming concepts, and available PLEs. This will ultimately improve the learning experiences of pupils in grades 4–6.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s10798-023-09860-8.

Funding Open access funding provided by Mälardalen University. The study was supported by Vetenskapsrådet.

Declarations

Conflict of interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

References

Bell, T., & Vahrenhold, J. (2018). CS unplugged—How is it used, and does it work? In H.-J. Böckenhauer, D. Komm, & W. E. Unger (Eds.), *Adventures between lower bounds and higher altitudes* (pp. 497–521). Springer Nature. https://doi.org/10.1007/978-3-319-98355-4_29

Bjursten, E.-L., Nilsson, T., & Gumaelius, L. (2023). Computer programming in primary schools: Swedish Technology Teachers' pedagogical strategies. *International Journal of Technology and Design Education*. 33. https://doi.org/10.1007/s10798-022-09786-7

Braun, V., & Clarke, V. (2006). Using thematic analysis in psychology. *Qualitative Research in Psychology, 3*(2), 77–101.

Bryman, A. (2014). *Samhällsvetenskapliga metoder [Social Research Methods]* (6th ed.). Liber AB.

Cheah, C. S. (2020). Factors contributing to the difficulties in teaching and learning of computer programming: A literature review. *Contemporary Educational Technology, 12*(2), ep272.

Doyle, A., Seery, N., Gumaelius, L., Canty, D., & Hartell, E. (2019). Reconceptualising PCK research in D&T education: Proposing a methodological framework to investigate enacted practice. *International Journal of Technology and Design Education, 29*(3), 473–491.

Ertmer, P. A., & Ottenbreit-Leftwich, A. T. (2010). Teacher technology change: How knowledge, confidence, beliefs, and culture intersect. *Journal of Research on Technology in Education, 42*(3), 255–284.

EU. (2006). Recommendation of the European parliament and of the council of 18 December 2006 on key competences for lifelong learning. https://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=OJ:L:2006:394:0010:0018:en:PDF

Evripidou, S., Doitsidis, L., Tsinarakis, G., Zinonos, Z., & Chatzichristofis, S. A. (2022). Selecting a robotic platform for education. In: 2022 IEEE International conference on consumer electronics (ICCE).

Fessakis, G., Komis, V., Dimitracopoulou, A., & Prantsoudi, S. (2019). Overview of the computer programming learning environments for primary education. *Review of Science, Mathematics and ICT Education, 13*(1), 7–33.

Fessard, G., Wang, P., & Renna, I. (2019). Are there differences in learning gains when programming a tangible object or a simulation? In B. Scharlau, R. McDermott, A. Pears, & M. Sabin (Eds.), *Innovation and technology in computer science education, ITiCSE 2019* (pp. 78–84). Association for Computing Machinery, ACM. https://doi.org/10.1145/3304221.3319747

Finger, G., & Houguet, B. (2009). Insights into the intrinsic and extrinsic challenges for implementing technology education: Case studies of Queensland teachers. *International Journal of Technology and Design Education, 19*(3), 309–334.

Friedrichsen, P., Driel, J. H. V., & Abell, S. K. (2011). Taking a closer look at science teaching orientations. *Science Education, 95*(2), 358–376.

Gess-Newsome, J. (2015). A model of teacher professional knowledge and skill including PCK: Results of the thinking from the PCK Summit. In A. Berry, P. Friedrichsen, & J. Loughran (Eds.), *Re-examining pedagogical content knowledge in science education* (pp. 38–52). Routledge.

Humble, N. (2021). The use of programming tools in teaching and learning material by K-12 teachers. In: 20 th European conference on e-Learning, ECEL 2021, Berlin, Germany.

Kelleher, C., & Pausch, R. (2005). Lowering the barriers to programming: A taxonomy of programming environments and languages for novice programmers. *ACM Computing Surveys, 37*(2), 83–137.

Koehler, M. J., Mishra, P., & Cain, W. (2013). What is technological pedagogical content knowledge (TPACK)? *Journal of Education, 193*(3), 13–19.

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*, 103872. https://doi.org/10.1016/j.compedu.2020.103872

Loughran, J., Berry, A., & Mulhall, P. (2012). *Understanding and developing scienceteachers' pedagogical content knowledge*. Springer Science & Business Media.

Makki, T. W., O'Neal, L. J., Cotten, S. R., & Rikard, R. (2018). When first-order barriers are high: A comparison of second-and third-order barriers to classroom computing integration. *Computers & Education, 120*, 90–97.

Meerbaum-Salant, O., Armoni, M., & Ben-Ari, M. (2011). Habits of programming in scratch. In G. Rößling, T. Naps, & C. Spannagel (Eds.), *Proceedings of the 16th annual joint conference on Innovation and technology in computer science education* (pp. 168–172). Association for Computing Machinery. https://doi.org/10.1145/1999747.1999796

Player-Koro, C., Bergviken Rensfeldt, A., & Selwyn, N. (2018). Selling tech to teachers: Education trade shows as policy events. *Journal of Education Policy, 33*(5), 682–703.

Price, T. W., & Barnes, T. (2015). Comparing textual and block interfaces in a novice programming environment. In B. Scharlau, R. McDermott, A. Pears, & M. Sabin (Eds.), *Innovation and technology in computer science education, ITiCSE 2019* (pp. 91–99). ACM: Association for Computing Machinery.

Przybylla, M., & Romeike, R. (2015). Key competences with physical computing. In T. Brinda, N. Reynolds, R. Romeike, & A. Schwill (Eds.), *KEYCIT 2014: key competencies in informatics and ICT* (Vol. 7, pp. 351–361).

Reichenberg, M. (2014). Predicting teachers' choice of teaching and learning materials: A survey study with Swedish teachers. *IARTEM e-Journal, 6*(2), 71–93.

Rich, P. J., Bartholomew, S., Daniel, D., Dinsmoor, K., Nielsen, M., Reynolds, C., Swanson, M., Winward, E., & Yauney, J. (2022). Trends in tools used to teach computational thinking through elementary coding. *Journal of Research on Technology in Education*. https://doi.org/10.1080/15391523.2022.2121345

Robson, C., & McCartan, K. (2016). *Real world research*. John Wiley & Sons.

Rohaan, E. J., Taconis, R., & Jochems, W. M. (2011). Exploring the underlying components of primary school teachers' pedagogical content knowledge for technology education. *Eurasia Journal of Mathematics, Science and Technology Education, 7*(4), 293–304.

Saeli, M., Perrenet, J., Jochems, M. G. W., & Zwaneveld, B. (2012). Programming: Teachers and pedagogical content knowledge in the Netherlands. *Informatics in Education, 11*(1), 81–114.

Saeli, M., Perrenet, J., M.G. Jochems, W., & Zwaneveld, B. (2010). *Portraying the pedagogical content knowledge of programming—The technical report*. http://teachingprogramming.esoe.nl/TechnicalReport/SPJZ/TechnicalReport.pdf.

Shulman, L. S. (1987). Knowledge and teaching: Foundations of the new reform. *Harvard Educational Review, 57*(1), 1–22.

Shulman, L. S. (2013). Those who understand: Knowledge growth in teaching. *Journal of Education, 193*(3), 1–11. https://doi.org/10.1177/002205741319300302

Skolinspektionen. (2014). *Teknik-gör det osynliga synligt [Technology - making the invisible visible]*. https://skolinspektionen.se/globalassets/02-beslut-rapporter-stat/granskningsrapporter/tkg/2014/teknik/kvalgr-teknik-slutrapport.pdf

Statens Skolverk. (2017a). *Få syn på digitaliseringen på grundskolenivå – Ett kommentarmaterial till läroplanerna förförskoleklass, fritidshem och grundskoleutbildning [Looking at digitisation at primary school level - A commentary on the curricula for pre-school, after-school and primary education]*. Retrieved from https://www.skolverket.se/publikationsserier/kommentarmaterial/2017/fa-syn-pa-digitaliseringen-pa-grundskoleniva

Statens Skolverk. (2017b). *Läroplan för grundskolan, förskoleklassen och fritidshemmet 2011 Reviderad 2017 [Curriculum for the compulsory school, preschool class and school-age educare 2011 (revised 2017)]* https://www.skolverket.se/publikationsserier/styrdokument/2017/laroplan-for-grundskolan-forskoleklassen-och-fritidshemmet-2011-reviderad-2017?id=3813

Statens Skolverk. (2018). *Curriculum for the compulsory school, preschool class and school-age educare*. Retrieved from https://www.skolverket.se/publikationsserier/styrdokument/2018/curriculum-for-the-compulsory-school-preschool-class-and-school-age-educare-revised-2018

Statens Skolverk. (2019). *Digital kompetens i förskola, skola och vuxenutbildning 2018 [Digital competence in preschool, compulsory school and adult education]* (Dnr: 2018:1292). https://www.skolverket.se/publikationsserier/rapporter/2019/digital-kompetens-i-forskola-skola-och-vuxenutbildning

Statens Skolverk. (2022). *Läroplan för grundskolan, förskoleklassen och fritidshemmet 2022 [Curriculum for the compulsory school, preschool class and school-age educare 2022]*

The Office for Standards in Education, C. s. S. a. S. (2013). *National curriculum in England: computing programmes of study*. Retrieved from https://www.gov.uk/government/publications/national-curriculum-in-england-computing-programmes-of-study/national-curriculum-in-england-computing-programmes-of-study?fbclid=IwAR3GlrQ0yKy3fiUKrKbq35gQToL2ldjceNw6sPn4mBWbMIC6N82CrXTbCI8#key-stage-2

Utbildningsdepartementet. (2002). *Delrapport från arbetsgruppen för en ny nationell it-strategi för skolan [Interim report from the working group for a new national IT strategy for schools]*. Retrieved from https://www.regeringen.se/49b71a/contentassets/45792cac59bf422ba8109bb1261dadf4/nasta-steg-delrapport-fran-arbetsgruppen-for-en-ny-nationell-it-strategi-for-skolan

Vinnervik, P. (2020). Implementing programming in school mathematics and technology: Teachers' intrinsic and extrinsic challenges. *International Journal of Technology and Design Education, 32*, 213–242. https://doi.org/10.1007/s10798-020-09602-0

Vinnervik, P. (2021). *När lärare formar ett nytt ämnesinnehåll - Intentioner, förutsättningar och utmaningar med att införa programmering i skolan [Teachers as curriculum makers : intentions, settings and challenges associated with integrating programming into schools]* [Doctorial dissertation, Umeå Universitet]. http://www.diva-portal.org/smash/record.jsf?pid=diva2%3A1589572&dswid=5480

Weintrop, D., & Wilensky, U. (2015). To block or not to block, that is the question: Students' perceptions of blocks-based programming. In O. Sejer Iversen, L. Elbæk, & B. Stjerne Thomsen (Eds.), *14th international conference on interaction design and children, IDC 2015* (pp. 199–208). Association for Computing Machinery. https://doi.org/10.1145/2771839.2771860

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), 235–254. https://doi.org/10.1080/08993408.2016.1257418

Zhang, X., Zhang, C., Stafford, T. F., & Zhang, P. (2013). Teaching introductory programming to IS students: The impact of teaching approaches on learning performance. *Journal of Information Systems Education, 24*(2), 147–155.

Authors and Affiliations

Eva-Lena Bjursten[1] · Tor Nilsson[1] · Gunnar Jonsson[2]

✉ Eva-Lena Bjursten
eva-lena.bjursten@mdu.se

Tor Nilsson
tor.nilsson@mdu.se

Gunnar Jonsson
gunnar.jonsson@mdu.se

[1] School of Education, Culture and Communication, Mälardalen University, Eskilstuna, Sweden

[2] School of Education, Culture and Communication, Mälardalen University, Västerås, Sweden