Early Childhood Education Journal
https://doi.org/10.1007/s10643-019-00976-y

Investigating Preschool Educators' Implementation of Computer Programming in Their Teaching Practice

Anna Otterborn[1] · **Konrad J. Schönborn**[2] · **Magnus Hultén**[1]

Abstract
Modern preschool education is seen as an essential foundation for nurturing children's digital literacy. Early childhood education environments have witnessed increased emphasis on integrating programming activities in combination with digital tablets. However, little is known about how preschool teachers implement programming as part of pedagogical strategies during practice. In Sweden, although there is a mandate to develop children's understanding of the digital world, programming is not formally mentioned in the revised preschool curriculum. This study systematically investigates how Swedish preschool teachers implement programming activities in their teaching practice. Data was collected through a national online survey (n = 199). Findings revealed a range of apps and resources used in combination with tablets, where activity integration takes place as unplugged programming, digital programming, or as a combination of the former. Teachers markedly attributed intended learning goals around programming to tenets of computational thinking and "twenty-first century skills". Moreover, programming was often actively linked to learning in other domains such as science, technology, mathematics, and language, approaches that show traditional Swedish preschool teaching practices being recontextualized in terms of programming. Based on the reported findings that provide insight into the implementation of programming in preschools, a logical future research avenue lies in exploring the documented programming activities from the perspective of the children.

Keywords Programming · Digital tablets · Swedish preschool · Technology education · Science education · iPads

Introduction

Implementation of computer programming in school teaching continues to grow internationally. The last decade has seen major attention directed towards the integration of programming activities in preschool classrooms, where notions of "digital literacy" and/or "computational thinking" form part of many education policy agendas. This trend is due both to political emphasis placed on the need for digitally literate citizens as well as on the necessity to adapt educational practices in a society that is becoming increasingly dependent on digital technologies (Howland et al. 2018).

The advent of more intuitive and tangible educational technology means that even very young children can gain access to programming—the child can input commands into a device without any knowledge of programming syntax or any prerequisite reading literacy (Ching et al. 2018; Gomes et al. 2018). Bers (2018a) believes that it is of utmost importance that preschool children are exposed to working with digital tools and programming activities. In an example of the increased interest in programming in early childhood education, Bers (2018b) has reported that as at early 2018, the *ScratchJr* visual programming language (aimed for children 5–7 years of age) had been downloaded 9.5 million times. The same study by Bers (2018b) shows that while ScratchJr users are predominantly located in North America and Britain, the programming language is most popular in Sweden (relative to country population size).

Implementation of programming activities at the preschool level often involves combining an app with physical artefacts such as robots. For example, Bers et al. (2014) investigated how 4–6-year-old children from three different kindergartens used a hybrid/graphical computer language (CHERP) to program a robotic vehicle that they built themselves. The results showed that the children

✉ Anna Otterborn
anna.otterborn@liu.se

1 Department of Social and Welfare Studies (ISV), Linköping University, 60174 Norrköping, Sweden

2 Department of Science and Technology (ITN), Linköping University, 60174 Norrköping, Sweden

Published online: 14 September 2019

worked collaboratively and were able to acquire and use various computational thinking abilities associated with debugging, correspondence, sequencing and control flow. Falloon (2016) suggests that implementing basic coding with the help of software resources such as ScratchJr can be a valuable teaching tool that provides students with opportunities to forge a diverse set of problem-solving skills. On this matter, an intervention study by Fessakis et al. (2013) showed that while working with *Logo*, children aged 5–6 years old developed social skills, problem solving abilities, and mathematical concepts. Furthermore, a robot programming teaching intervention for 3–5-year-olds developed by Palmér (2017) was found to benefit children's spatial and counting ability, and their understanding of symbols (e.g., arrows). Gomes et al. (2018) suggest that as part of a holistic approach to teaching computational thinking, combining digital and unplugged activities can improve children's understanding of programming concepts.

This study took place when the former national preschool curriculum was still in effect, while a new revision was in preparation (and is implemented today). While the former Swedish preschool curriculum (Swedish National Agency for Education 2010) did not feature digital literacy, the newly revised preschool curriculum has "digital literacy" formally included, and also prescribes the use of "digital tools" (Swedish National Agency for Education 2018, pp. 9, 14, 15, and 19). However, programming is not specifically mentioned in the new curriculum, even though the document emphasizes that children should be given the opportunity to develop an understanding of the digital world (Swedish National Agency for Education 2018). Despite programming not being formally specified, our earlier work (Otterborn et al. 2019) has shown that many preschool teachers have already begun to implement programming in Swedish preschools in earnest through various self-propelled initiatives that make use of digital tablets in combination with a range of applications, activities and artefacts. Albeit so, teachers have also expressed a lack of knowledge and skills related to programming. Our previous research also affirms that preschool teachers desire clearer accompanying curriculum directives (Otterborn et al. 2019) and seek more support for how to optimally use digital tablets in educational activities (also see Kjällander and Frankenberg 2018).

Increasing international demands to integrate digital literacy into preschool education poses both opportunities and challenges for preschool teachers (Ching et al. 2018; Neumann and Neumann 2014; Palmér 2015; Strawhacker et al. 2018; Yelland and Masters 2007). In pursuit of gaining insight into these developments, the aim of this study is to survey what is currently occurring in preschools with respect to how Swedish preschool teachers implement programming activities during teaching practice.

Theoretical Perspectives for Considering Programming in Childhood Education

In the 1960s, the USA made major investments in computers and associated educational software (Emanuel 2009). Seymour Papert, a pioneer of computer science and education, was one of the co-creators of Logo, the first programming language developed for children. In applying the metaphor of "low floors and high ceilings", Papert considered it important that a programming language enable simple and advanced exercises, which could cater to both beginners and more advanced users (Resnick et al. 2009). Papert (1980) aimed for a child's knowledge and abilities developed through Logo to be potentially transferrable to other learning scenarios. Logo also laid the foundation for the development of subsequent programming languages such as ScratchJr (Kazakoff et al. 2013).

Contemporary discussions around programming in childhood education often draw from Papert's (1980) ideas of procedural thinking, and as a result of Wing's (2006) influential paper, have manifested in the idea of *computational thinking* (CT). CT is framed as central in relation to early childhood programming education (Howland et al. 2018). According to Wing (2006), CT promotes thinking like a computer scientist when having to solve problems and is deemed a universal competence that is important for all children in the twenty-first century. The concept of CT has been widely discussed, and different definitions and variations of the concept have been proposed (e.g., Barefoot 2019; Bers et al. 2014). In this light, Mannila (2017) states that "computational thinking is usually defined as a collection of concepts and practices that help solve problems and generate new creations with the computer" (p. 87). Furthermore, Ching et al. (2018) stress that CT can be applied across multiple problem-solving contexts:

> […] a broad problem-solving framework involving skills, processes, and approaches to solve problems, and programming as a key practice for supporting and cultivating the cognitive tasks involved in computational thinking (Ching et al. 2018, p. 564).

Regarding CT skills related to broad problem-solving strategies, the Barefoot (2019) programme in Britain frames computational thinking as comprising six concepts, namely logic, algorithms, decomposition, patterns, abstraction, and evaluation, in combination with the following five approaches: tinkering, creating, debugging, persevering and collaboration. In an additional synthesis, Bers et al. (2014) highlight the following skills as specifically important in relation to preschool education:

> Problem representation; systematicity in generating and implementing solutions; exploring multiple pos-

sible solutions; problem-solving on multiple levels – from approaching the overall challenge to "debugging" or trouble-shooting specific difficulties with a given solution's implementation; productive attitudes toward "failure" and misconceptions uncovered along the route to a successful project; and strategies for approaching open-ended and often difficult problems (Bers et al. 2014, p. 146).

In addition to the position put forward above, various concepts, such as "digital fluency" are closely related to CT skills, and concern competencies that include the ability to use different computer devices for learning, and to use technology to reach desired outcomes or problem solutions (e.g., Resnick et al. 2009).

Aim of the Study

Research on computational thinking and the use of digital (and non-digital) tools in supporting the teaching of programming is still developing at the preschool level (Howland et al. 2018). More specifically, little is known about what exact activities and systematic strategies are being adopted during preschool educational practice (Palmér 2017). The aim of this study is to investigate Swedish preschool teachers' views on the content, methods and purposes associated with implementing programming activities in their teaching practice. We believe that gaining detailed insight into what activities are currently underway, and how they are implemented at the practice level can be used to inform future curriculum design to support this area of pedagogical intervention.

Methods

Study Context, Participants and Design

The study consisted of administering an online survey to Swedish preschool teachers. The online survey was designed to generate information about preschool teachers' views, use, and implementation of programming activities.

Data Collection

Design of the currently reported survey draws on the results of a preceding survey study conducted by the authors (Otterborn et al. 2019), where surveyed teachers were found to emphasise programming as one of the salient activities related to the use of digital tablets in preschools. In this regard, since programming is mentioned in the revised elementary school curriculum (Swedish National Agency for Education 2010) but not formally operationalised in the new preschool curriculum, we propose that preschool teachers were inspired to integrate programming activities in their own preschool teaching practice. Our earlier results also showed that the use of digital tools was a core part of traditional preschool activities that incorporated a strong emphasis toward programming applications. This former result highlighted the fact that preschool educators are already very involved in using both digital and non-digital means to foster programming activities, which stimulated the design of the questions reported here. In doing so, in the preceding survey study, the teachers identified various programming apps used in their teaching and emphasised the importance of programming skills at the preschool level. Hence, design of the currently reported survey evolved from, and exclusively aimed to expand upon and home in on, these previously revealed aspects of programming. In assessing the face validity (e.g., Logue and Harvey 2009) of fifteen developed survey items, two preschool teachers piloted the items and stipulated that they found them comprehensible and relevant.

The fifteen online survey questions first gathered demographic information on teachers' gender, age, pedagogical role, and their preschool location. The next group of questions obtained information on the regularity of programming activities during teaching (i.e., daily, weekly, monthly, varying, or never), teachers' views on potential advantages of programming in different subject areas (e.g., natural science, technology, mathematics, language), as well as the potential cognitive and social benefits of programming activities (i.e., problem solving, critical thinking, cooperation, strategic thinking, environmental understanding, confidence building, or none of these). Teachers were also asked who served as the initiator of programming activities (e.g., the educator themselves or the preschool manager), whether there was a common strategy for programming in their respective preschool or preschool division, whether programming activities were also integrated without digital tools (unplugged programming), and what specific apps (e.g., Run Marco, Lightbot Jr, Coda Game, Loopimal, Osmo, Blue-Bot, Bee-Bot, or others) were used when programming with digital tablets. The survey also asked teachers to state whether they thought females and males used tablets in different ways. The online survey concluded with an open question asking respondents to describe actual examples of how they implement programming during preschool practice.

The survey was activated from February 2018 to June 2018. The survey link was sent to approximately 500 preschool teacher email addresses across Sweden. While active, the survey was answered by 199 respondents (wherein 105 answered the open question on how programming was implemented during practice), which represents the survey data analysed in this study.

Data Analysis

The corpus of 199 teacher survey responses was treated in the following three-step procedure. Firstly, responses to the closed-ended items were tabulated and quantified in terms of relative proportion (%) of respective item response options. This deductive process identified teacher demographics, preschool location, frequency of programming activities, perceived initiator of programming activities, existence of a common strategy for programming in the preschool, which programming apps were used, what subject areas were viewed as most conducive to potential benefits of programming, whether teachers also initiated unplugged programming activities, whether females and males differed in tablet use, and lastly, perceived cognitive and social benefits of programming. Secondly, the open-ended item (n = 105) was subjected to a thematic content analysis (Mayring 2014; Miles et al. 2014) with the objective to iteratively identify any theme and sub-theme structure in the data corpus. This inductive process discerned emergent themes around how teachers integrate real programming activities during preschool education.

Results

The findings of the study are reported as follows. First, the demographic features of the survey respondents, together with a synthesis of preschool educators' views and use of programming through various tablet apps are provided. Second, categories of implemented preschool programming activities in relation to educators' intended learning goals that emerged from the survey are distilled.

Demographic Characteristics and Features of the Survey Respondents

Demographics (gender and age) and context features (preschool location, pedagogical role) of the respondent sample is summarised in Table 1.

The demographic data reveals that responses were obtained from across Sweden, with almost all (92.4%) of the respondents being female (Table 1). The survey revealed a response distribution across all representative ages of preschool educators. The 9.6% proportion of respondents that selected their pedagogical role as "Other" included stated roles such as part-time pedagogues and cultural educators (Table 1).

Table 1 Demographic characteristics and context features of survey respondents (n = 199)

Demographic and context features	Proportion of sample (%)
Gender	
Female	92.4
Male	6.1
Other	1.5
Age	
20–30	16.8
31–40	25.0
41–50	27.6
51–60	23.5
≥ 61	7.1
Pedagogical role	
Preschool teacher	84.3
Childminder	6.1
Other	9.6
Preschool location in Sweden	
Northern Sweden	4.6
Central Sweden	31.0
Southern Sweden	64.4

Preschool Educators' Use and Views of Programming in a Preschool Context

Educators' use and views of programming (subject areas where programming may be advantageous, student abilities that may benefit, how often program activities occur, initiator of programming activities, whether there is a common strategy for implementing programming at the preschool, whether programming activities without digital tools occurs, any perceived gender differences, and the apps used for programming) are summarised in Table 2.

As presented in Table 2, preschool teachers felt that programming could benefit the learning of various subject areas, with the largest proportion of teachers interpreting programming to be most advantageous for learning technology (92.4%) and mathematics (93.9%). In addition, respondents viewed the abilities of problem solving (89.3%), cooperation (82.2%), and strategic thinking (81.2%) to be promoted most through programming activities. In terms of the prevalence of programming activities, 43.3% of the respondents stated that activities occurred with varying regularity, and almost half (47.2%) of the preschool teachers said that the activities stemmed from their own initiative, where 51.3% of the teachers suggested that they also made use of unplugged programming activities (approximately two-thirds of the preschool teachers indicated that there was no common school or division strategy for implementing programming activities). In this regard, the 10.9% proportion of answers captured in "Other" corresponded to initiatives directed at the municipal level. About one quarter of the survey sample felt that there was a difference (with three quarters perceiving no difference) in the way males and females interacted with tablets. Lastly, while most of the respondents indicated Blue-Bot (52.3%) and Bee-Bot (46.4%) to be the most commonly

Table 2 Preschool educators' views and implementation of programming in preschool (n = 199)

Educators' views and implementation of programming	Proportion of sample (%)
Programming can be used as an advantage in this area	
Natural science	62.9
Technology	92.4
Mathematics	93.9
Language	79.2
None of these options	1.0
This ability benefits from programming	
Problem solving	89.3
Critical thinking	40.6
Cooperation	82.2
Strategic thinking	81.2
Environmental understanding	39.1
Confidence in one's ability	76.1
None of these options	1.5
When programming takes place	
Every day	4.6
Every week	13.3
Every month	6.1
Irregularly—varies in different periods	43.3
Never	32.7
Initiator of programming	
Educator's initiative	47.2
Initiative of preschool manager	12.4
We do not program at our preschool	29.5
Other	10.9
Common strategy for programming at the division	
Yes	32.8
No	67.2
Common strategy for programming at the entire preschool	
Yes	18.9
No	63.8
Do not know/undecided	17.3
Programming without digital tools	
Yes	51.3
No	48.7
Boys and girls use touch tablets in different ways	
Yes	23.4
No	76.6
Software apps used for programming	
Run Marco	2.6
Lightbot Jr	18.3
Coda Game	3.9
Loopimal	9.2
Osmo	14.4
Blue-Bot	52.3
Bee-Bot	46.4
Other	32.7

[a]In some of the survey items more than one option could be selected

[b]A typical Swedish preschool contains approximately four divisions

used apps in preschool programming activities, apps that included ScratchJr, Scottie Go, Go for Dash, Code carts and A.L.E.X were identified in the "Other" category.

Preschool Teachers' Implementation of Programming Activities During School Practice

Upon analysis of received open-ended survey responses (n = 105), preschool teachers' implementation of three overall types of programming activities emerged, namely unplugged programming (n = 9), digital programming (n = 33) or implementation of the former in combination (n = 31). Out of the remaining 32 responses, 14 respondents stated they did not work with programming and 18 responses were too brief to categorize. The emergent three approaches are described below, as well as the intended learning goals that teachers mentioned in conjunction with the activities.

Unplugged Programming

Unplugged programming is defined as the teaching of programming concepts without a computer device (e.g., Faber et al. 2017), and is often implemented through bodily activity or using different objects and apparatus. Various examples of unplugged programming activities were described that included creating patterns with beads, recognizing patterns in pictures, performing yoga and gymnastics while following instructions (both on paper via symbols and through imitation), and giving movement instructions to children (dressed as robots) in order to mimic "programming a robot".

The survey showed that nine preschool teachers use unplugged programming as an introduction to digital programming. The argument the teachers gave for implementing this approach is that unplugged programming provides the students with more concrete experiences that can later facilitate the digital programming work that follows. Other examples of unplugged programming activities included children creating and following algorithms/instructions using verbal instructions, symbols, images, and/or photos. In another example, the children were photographed when performing different dance moves, and these photos were then used by the children to create their own choreography, which they later performed together. Other examples of unplugged programming that were stated included working with pattern recognition and bead sorting tasks, where children attempt to repeat the same pattern as provided on a template or continue to build an already developed pattern.

Digital Programming

The survey revealed two distinct types of digital programming, namely programming directly on the digital tablet screen using various apps (digital programming), and programming with the help of tangible manipulatives such as Blue-Bot and other robots. The latter often also involves design and construction of other artefacts such as using different materials to create courses and obstacles for the robots. We refer to this type of programming as physically extended digital programming. When tangible robots are used, these can be programmed either from the digital tablet screen using various apps, or with the help of block programming that uses a Blue-Bot Tactile Reader (a stand-alone device that can control the Blue-Bot), for example. In terms of physically extended digital programming using Blue-Bot and other robots, the survey also showed it commonplace to use ready-made and/or self-made carpets with squares (of correct sizes for the robot). Blue-Bot or other robots can also be "dressed up" to represent different characters and animals that are programmed to move to different location goals. In order to reach the goal, the robots are required to navigate different obstacles that the children have built. Other examples of using tangible robots at preschools included attaching a pen to a Blue-Bot so that children can program the robot to draw different figures, such as geometric shapes.

Another identified example of physically extended digital programming is when preschool teachers read a story to the children. In these activities, pictures from the story are placed on the floor beneath a transparent mat that is divided into squares. The children then recount the story while programming Blue-Bot to move to the different pictures in the correct order of the narrative. In physically extended digital programming, the children also design and build tracks and obstacles from different materials such as bricks, clay, Lego and Kapla rods. The children also use their own bodies as bridges.

Unplugged and Digital Programming Implemented in Combination

The survey showed that unplugged and digital programming activities also occur in combination. In these cases, while some students program Blue-Bot on the mat using a digital tablet, a connected projector allows everyone in the room to also view what is occurring. Some children then "program" each other's bodies inspired by the unveiled digital activities. Thirty-one preschool teachers explained that they implement various types of programming during the same learning situation. They often commence with unplugged programming to generate a concrete experience for the students, and then proceed to a digital programming activity.

The survey responses also emphasized that integrating programming activities in project work makes the nature of programming more meaningful for the children. One stated example of project work in relation to unplugged programming was health. A further example of project work in relation to unplugged and digital programming was a project about the town where one of the preschools was located. In this case, the children first carefully explored the local city environment through walks and by photographing landmarks with the digital tablet. The children then recreated the immediate environment and programmed Blue-Bot to navigate to specific locations in the recreated city and then return "home" to the preschool again. The children also discussed the construction and strength of materials during the same task while using plastic tape, bricks and clay to build represented city bridges.

Learning Goals Associated with Implementing the Programming Activities

Three different clusters of learning goals associated with the collective programming activities emerged from the survey responses. These were categorised as follows:

Overarching learning goals concerning a digital society

- Understand the digital world.
- Understand that a human being controls the computer/robot.
- Obtain a concrete understanding of what happens in programming.

Programming related skills and learning goals

- Generate an understanding of symbols and how they can be used (in the case of unplugged programming, this relates to symbols indicating how to move the body, or to arrows or symbolic representations in certain positions).
- Develop, provide, interpret and perform instructions that include following a specific order/sequence, or perform stepwise procedures.
- Break down a task into smaller task components.
- Debug and attempt the task/procedure again when required.

General skills and abilities

- Trust in one's own ability.
- Develop new ideas/inventions, create and design.
- Understand that there are different ways to solve a problem and reach a solution.
- Develop collaborative, motor and social skills.

Discussion and Implications

A wide range of programming apps are becoming increasingly available for use in preschool education (Ching et al. 2018; Mannila 2017). Integrating symbolic language in combination with user-friendly digital tablets most likely represents the extent to which programming is currently being implemented in Swedish preschools, and perhaps internationally. In Sweden, swift developments in the active integration of programming activities in preschool may also have been catalysed by the formal inclusion of programming in the new Swedish elementary school curriculum (Otterborn et al. 2019). The objective of this study was to investigate preschool teachers' views of programming and how they implement programming during their preschool practice. Most previous research on programming in preschool has concerned studying programming interventions that have been designed by the investigating researchers (e.g., Bers et al. 2014; Fessakis et al. 2013; Palmér 2017). In contrast with (yet adding an additional lens to) this approach, the present study has focused on discovering what preschool teachers do when integrating programming in practice. In addition, while much previous research has concentrated on children of 5 years of age and older (e.g., Palmér 2017), this study sheds light on early childhood programming activities occurring with children younger than six years old. In this regard, our results show that programming is often integrated with multiple *content* areas such as project work, and linked to various conceptual domains (e.g., science, technology, mathematics, language). In addition, preschool teachers use various *methods* to embed programming activities that comprise unplugged, digital programming and physically extended digital programming in Swedish preschools. Furthermore, preschool teachers' integration of programming activities is interwoven with distinct *purposes* that include cooperation, scaffolding children's (creative) problem-solving and building trust in children's own ability. Moreover, as part of pursuing these envisaged purposes of programming, teachers are aware of various skills, abilities and learning outcomes related to computational thinking, and aim to actively integrate these when developing and implementing programming activities in practice.

Although the reported work provides insight into how programming is implemented in a Swedish preschool context, given the general character of this domain, the findings may also be of potential international relevance (cf. Nilsen 2018). Furthermore, the approaches and activities revealed by the survey indicate that the implementation of programming endeavours are not localised to certain areas in the country but are widespread. On this score, it is noteworthy that not all preschools integrate programming, as indicated by a third of survey respondents suggesting that no programming activities take place in practice.

Integrating Programming in Swedish Preschools—General Findings

Overall, the salient general findings in response to the research aim are as follows:

- Different types of programming embedded in unplugged, digital programming and physically extended digital programming activities are present in Swedish preschools.
- Programming is often integrated with other related themes, project work, and linked to other conceptual domains (e.g., science, technology, mathematics, language).
- Cooperation with one-another, scaffolding children's (creative) problem-solving and building trust in children's own ability is very much at the forefront of preschool teachers' integration of programming activities.
- Teachers are aware of various skills, abilities and learning outcomes related to computational thinking, and aim to actively integrate these when developing and implementing programming activities.

The findings demonstrated two main approaches toward programming in preschools, namely unplugged programming (without digital tools) and digital programming (either exclusively with digital tools, or in combination with physical objects). Concerning *unplugged programming*, the most commonly exposed activity was allowing one child (or pedagogue) to take on the role of a "robot", while another child provided instructions to move and navigate the robot. Unplugged programming activities also included sorting objects such as bead tasks, with the intention for children to perceive patterns and follow sorting instructions (sequencing). The latter activities have a long pedigree in Swedish preschool education and indicate that traditional teaching practices are becoming dynamically recontextualized in terms of programming concepts. When it comes to *digital programming*, the study elucidated several methods adopted by teachers. One pronounced example was programming with tangible manipulatives such as Blue-Bot and other robots (more than half of the survey respondents stated the use of Blue-Bot in programming). Related activities involved programming directly on the digital tablet screen with various apps, using an app to program a Blue-Bot, or using the Blue-Bot Tactile Reader with block programming. Findings showed that teachers also *combined* unplugged and digital programming. In these cases, they often commenced with unplugged programming to provide a more concrete experience of programming for the children, and then progressed to digital programming activities (cf. Faber et al. 2017; Gomes et al. 2018; Mannila 2017). Unplugged and digital programming also occur simultaneously. For example, some students program Blue-Bot on the floor or with a tablet. Projection of the digital tablet interface allows all children to view what is happening on the digital screen, while other children program each other's bodies based on the projected activities.

Since not all teachers who stated that they use programming answered the open-ended question, it is not possible to link the quantitative results from the thematic analysis to how common different approaches were in our sample. Nevertheless, it is notable that out of the 73 answers that could be classified, combined approaches emerged almost as commonly as digital, while those that only used non digital approaches were clearly in the minority (n = 9). Further research could explore these categories to establish how common they are in practice.

Three different clusters of learning goals were discerned in the survey responses that could be linked to both unplugged and digital programming. These concerned the notion of a *digital society*, such as gaining an understanding of the digital world, *programming related skills* and lastly, *general skills* such as trust in one's own ability, creativity and social skills. Many of the pursued goals and skills overlap with the idea of "twenty-first century skills" that incorporate critical thinking, creativity, collaboration, communication, information and media literacy, technology literacy, initiative and productivity, and social skills, and that are also stressed in the current and former preschool curriculum. This study suggests that preschool teachers view programming as one vehicle through which many of these skills can be developed and fostered. Moreover, given that specific programming related skills and learning goals are not explicitly stated in neither the old nor the new curriculum, the findings reported in this study can be seen as a sign that teachers themselves are developing more specified learning goals in this area based on their own views. The findings also demonstrate a resonance between the skills highlighted by the preschool teachers' programming activities in our study and skills defined in relation to CT in the literature (Bers et al. 2014; Ching et al. 2018). Indeed, the data suggests that solid work in implementing programming at preschools is in operation around Sweden. More than two-thirds of the survey respondents stated that they carry out programming activities at their respective schools, with 13% of respondents implementing activities on a weekly basis. Albeit so, about two-thirds of the preschool teachers reported that there is no common strategy for implementing programming within their own division, a finding further emphasised by almost half of the respondents stating that they implement programming activities based on their own initiative.

Challenges and Implications for Teaching Practice and Curriculum Development

Apart from garnering what activities, processes and learning goals are associated with programming in preschool, this study has also illuminated certain challenges and possible ambiguities associated with the implementation of programming. In this regard, we highlight the following two emergent observations that are further unpacked in relation to implications for teaching practice and curriculum development:

- Several subjects were linked to programming activities as a basis for improving learning in these areas and were largely influenced by thematic/project work approaches and the notion of computational thinking.
- Programming activities are almost exclusively contextualized and framed in relation to robots.

This study has shown that preschool teachers place significant emphasis on framing and describing learning outcomes around programming in terms of computational thinking skills and more generic twenty-first century skills. Our findings point to preschool teachers' strong belief in programming as a potential new lingua franca, a means to swiftly foster generic skills. There are clearly sound arguments in continuing to implement programming in preschool, but how does this pedagogically align with already existing subject content and activities in preschool? The results of this study already show signs of traditional teaching practices being recontextualized in terms of programming, but more insight is needed into what children learn through these activities. Also—at present, programming is almost always contextualized in relation to robot programming tasks. How does this influence preschool children's constructions about their views of the digital world? In response, we suggest that a next step of empirical inquiry would be to study programming activities from the perspective of the children.

Finally, in addressing the limitations of this study, we did not ascertain preschool teachers' previous experience or formal training in digital literacy or programming, nor the impact of teachers' age ranges on their views of how programming is implemented in practice, which in itself would constitute a valid line of future research inquiry. It is notable that the notion of training emerged in the open-ended responses, where some teachers mentioned ongoing, future, or previous training in this area. Given that some responses also asserted that preschool was an unsuitable venue for implementing digital tools, and given that the use of digital tools is common place in modern preschool education, future empirical investigation of such opposition toward this technological emergence is also of high relevance to early childhood education.

Acknowledgements We thank the preschool educators for participating in the study. Open access funding provided by Linköping University.

Compliance with Ethical Standards

Conflict of interest The authors declare that they have no conflict of interest.

Ethical Approval All procedures performed in this study involving human participants were in accordance with the ethical standards of the Swedish Research Council (Vetenskapsrådet 2011). The introductory page of the online survey questionnaire informed potential respondents about the aim of the study, followed by the statement, "By ticking the box below, I agree to participate in this survey and allow the documented material to be used in the research. The material will be treated confidentially and in accordance with the research ethics principles described by the Swedish Research Council". Respondents provided consent to participate in the survey when they ticked the box beside the statement that read, "I wish to participate in the survey".

References

Barefoot. (2019). Retrieved January 9, 2019 from https://www.barefootcomputing.org.

Bers, M. U. (2018a). *Coding as a playground: Programming and computational thinking in the early childhood classroom*. London: Routledge.

Bers, M. U. (2018b). Coding and computational thinking in early childhood: The impact of ScratchJr in Europe. *European Journal of STEM Education, 3*(3), 08. https://doi.org/10.20897/ejsteme/3868.

Bers, M. U., Flannery, L., Kazakoff, E. R., & Sullivan, A. (2014). Computational thinking and tinkering: Exploration of an early childhood robotics curriculum. *Computers & Education, 72,* 145–157.

Ching, Y. H., Hsu, Y. C., & Baldwin, S. (2018). Developing computational thinking with educational technologies for young learners. *TechTrends, 62*(6), 563–573.

Emanuel, M. (Ed.). (2009). *Datorn i skolan: Skolöverstyrelsens och andra aktörers insatser 1970- och 80-tal: Transkript av ett vittnesseminarium vid Tekniska museet i Stockholm den 30 oktober 2008*. Stockholm: Royal School of Technology.

Faber, H. H., Wierdsma, M. D. M., Doornbos, R. P., van der Ven, J. S., & de Vette, K. (2017). Teaching computational thinking to primary school students via unplugged programming lessons. *Journal of the European Teacher Education Network, 12,* 13–24.

Falloon, G. (2016). An analysis of young students' thinking when completing basic coding tasks using Scratch Jnr. On the iPad. *Journal of Computer Assisted Learning, 32*(6), 576–593.

Fessakis, G., Gouli, E., & Mavroudi, E. (2013). Problem solving by 5–6 years old kindergarten children in a computer programming environment: A case study. *Computers & Education, 63,* 87–97.

Gomes, T. C., Falcão, T. P., & Tedesco, P. C. D. A. R. (2018). Exploring an approach based on digital games for teaching programming concepts to young children. *International Journal of Child-Computer Interaction, 16,* 77–84. https://doi.org/10.1016/j.ijcci.2017.12.005.

Howland, K., Good, J., Robertson, J., & Manches, A. (2018). Editorial—Special issue on computational thinking and coding in childhood. *International Journal of Child-Computer Interaction*. https://doi.org/10.1016/j.ijcci.2018.11.001.

Kazakoff, E. R., Sullivan, A., & Bers, M. U. (2013). The effect of a classroom-based intensive robotics and programming workshop on sequencing ability in early childhood. *Early Childhood Education Journal, 41*(4), 245–255.

Kjällander, S., & Frankenberg, S. J. (2018). How to design a digital individual learning RCT-study in the context of the Swedish preschool: experiences from a pilot-study. *International Journal of Research & Method in Education, 41*(4), 433–446.

Logue, M. E., & Harvey, H. (2009). Preschool teachers' views of active play. *Journal of Research in Childhood Education, 24*(1), 32–49.

Mannila, L. (2017). *Att undervisa i programmering i skolan – varför, vad och hur?*. Lund: Studentlitteratur AB.

Mayring, P. (2014). *Qualitative content analysis: Theoretical foundation, basic procedures and software solution*. Klagenfurt: Sage.

Miles, M. B., Huberman, A. M., & Saldaña, J. (2014). *Qualitative data analysis: A methods sourcebook* (3rd ed.). Thousand Oaks: Sage.

Neumann, M. M., & Neumann, D. L. (2014). Touch screen tablets and emergent literacy. *Early Childhood Education Journal, 42*(4), 231–239.

Nilsen, M. (2018). *Barns och lärares aktiviteter med datorplattor och appar i förskolan*. (Doctoral dissertation). Retrieved June 1, 2019 from https://gupea.ub.gu.se/bitstream/2077/57483/1/gupea_2077_57483_1.pdf (ISBN: 978-91-7346-976-0).

Otterborn, A., Schönborn, K., & Hultén, M. (2019). Surveying preschool teachers' use of digital tablets: General and technology education related findings. *International Journal of Technology and Design Education, 29*(4), 717–737.

Palmér, H. (2015). Using tablet computers in preschool: How does the design of applications influence participation, interaction and dialogues? *International Journal of Early Years Education, 23*(4), 365–381.

Palmér, H. (2017). Programming in preschool—with a focus on learning mathematics. *International Research in Early Childhood Education, 8*(1), 75–87.

Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. New York: Basic books.

Resnick, M., Maloney, J., Monroy-Hernandez, A., Rusk, N., Eastmond, E., Brennan, K., et al. (2009). Scratch: Programming for all. *Communications of the ACM, 52*(11), 60–68.

Strawhacker, A., Lee, M., & Bers, M. U. (2018). Teaching tools, teachers' rules: Exploring the impact of teaching styles on young children's programming knowledge in ScratchJr. *International Journal of Technology and Design Education, 28*(2), 347–376.

Swedish National Agency for Education. (2010). *Läroplan för förskolan Lpfö 98*. Stockholm: Skolverket.

Swedish National Agency for Education. (2018). *Läroplan för förskolan Lpfö 18*. Stockholm: Skolverket.

Vetenskapsrådet, (2011). *God forskningssed*. Stockholm: Vetenskapsrådet.

Wing, J. M. (2006). Computational thinking. *Communications of the ACM, 49*(3), 33–35.

Yelland, N., & Masters, J. (2007). Rethinking scaffolding in the information age. *Computers & Education, 48*(3), 362–382.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.