LUMAT
International Journal on Math, Science and Technology Education

Research article

Unplugged programming in an early primary technology classroom: What conditions for learning are created?

Astrid Berg[1], Cecilia Axell[1] and Inger Eriksson[2]

[1] Linköping University, Sweden
[2] Stockholm University, Sweden

Abstract: Although using computers is the most common strategy to teach programming, "unplugged programming" (UP) has gained in popularity within educational settings. UP refers to the act of programming without a computer. However, research on UP has mainly focused on upper primary, middle-and high school students. The limited number of studies on UP in early primary school are, in addition, mainly quantitative effect studies. In this case study, we focused on the UP-classroom practice in early primary school during one lesson in technology (grade 1, 6-7-year-old students). The programming content, or the object of learning, that was in focus during the lesson was students' capability to understand the idea of *sequencing commands*. The aim of the study was to explore what aspects of the object of learning emerge as critical in the UP classroom. Our analysis revealed that the students, to be able to understand the idea of sequencing commands, needed to discern several rather detailed aspects. Importantly, one can't take for granted that these are aspects they discern when observing or interacting with programmable artefacts out of school. Rather, the results imply that it is a content that needs to be explicitly dealt with in the primary technology classroom.

Keywords: unplugged programming, primary school, variation theory

Correspondence: astrid.berg@liu.se

1 Introduction

Today, society is increasingly dependent on and shaped by computer technology. In response to this ongoing development, countries across the world have introduced programming into their curricula. In Sweden, programming was introduced into the technology and mathematics syllabus in 2018 for school years 1 (age 7) to 12 (age 19) (Mannila & Heintz, 2023; Skolverket, 2024).

Although using computers is the most common strategy to teach programming, "unplugged programming" (UP) has gained in popularity within educational settings (Ahmed et al., 2020; Humble et al., 2019). UP refers to the act of programming without a computer (Aranda & Ferguson, 2018; Bell et al., 2009). The idea is to learn some basic programming principles using oral or written instructions for how to perform a task

Received 22 May 2025 / Accepted 2 September 2025 / Published 9 September 2025
Published by the University of Helsinki, Finland and LUMA Centre Finland.

systematically in several steps. Bergqvist (2021) describes it as a “type of programming activity that shifts focus from the ”coding” on a computer” (p. 21) to a more problem-solving stance. Other researchers suggest that UP can be a meaningful way to introduce programming particularly to students in early primary school (Bell & Vahrenhold, 2018; Brackmann et al., 2017; Faber et al., 2017; Wohl et al., 2015).

An obvious advantage of UP activities is that they provide teachers with a hands-on and practical way of teaching programming (Aranda & Ferguson, 2018). Research also indicates that UP is a promising instructional strategy for helping students to develop computational thinking (CT) skills (Battal et al., 2021; Chen et al., 2023). Importantly, UP activities were found to promote computational thinking skills among the youngest students to a greater extent than plugged-in programming (Sun et al., 2024). However, research on UP has mainly focused on upper primary, middle- and high school students, with the aim to foster students’ interest in computer science (Brackmann et al., 2017; Wohl et al., 2015). The limited number of studies on UP in early primary school are mainly quantitative effect studies exploring how UP develops students’ capabilities through various activities (del Olmo-Muñoz et al., 2020; Sun et al., 2024). Research focusing on classroom practice in early primary grades and the interaction between students, teacher and different tools is, in other words, very scarce. In their literature overview, Huang and Loi (2021) point out that:

> a major [research] gap is descriptions of how unplugged activities are facilitated … how are [teaching] materials taught, especially in authentic classroom settings? … How are teachers using unplugged strategies in schools …? (p. 100) We need to better understand the affordances of unplugged approaches relative to other teaching strategies and to know who benefits from these activities. (p. 105)

Therefore, it seems urgent to develop knowledge about what students who participate in a classroom practice framed by UP activities have the opportunity to learn.

With this in mind, we are interested in exploring an introductory UP lesson in an early primary technology classroom. The programming content, or the object of learning, in focus during the lesson is students’ capability to understand the idea of *sequencing of commands*. The aim of the study is to explore and discuss what aspects of the object of learning emerge as critical in the UP classroom. The concepts of critical aspects and objects of learning are central in variation theory, which is used in this article as a theoretical lens. “Critical aspects” refers to aspects of a particular object of learning that students need to discern in order to develop a more qualified way of knowing (Marton & Fung-Lo, 2007). The following research question specifies the aim:

> What aspects of the object of learning can be distinguished in the student-teacher actions during UP activities and which of these aspects can be identified as critical?

2 Background and previous research

In this article, we will relate to aspects of teaching and learning programming in early primary school. In the following sections, we firstly provide some background information on the Swedish technology curriculum in early primary school and what it means to understand the main idea of programming. The second section focuses on computational thinking in the context of UP. In the third section, we present prior research focusing on aspects of instructional approaches that seem to support the learning of programming in early primary school.

2.1 Programming as a technology content in the early primary curriculum – what does it mean?

In the technology subject in Sweden, the main purpose of teaching programming is to develop students' understanding of programming as a technology in their everyday life (Skolverket, 2024), that is, to make programming visible and intelligible. For such purposes it is, however, necessary to develop students' knowledge not only about programming, but also *in* programming. Thus, from the perspective of the technology subject, a basic ability to program is a tool for understanding the function, structure and use of everyday technological solutions. In grades 1–3, "controlling objects with programming" is part of the technology curriculum in Sweden (Skolverket, 2024).

In this study, the programming content that is in focus in the classroom concerns programming as the ordering of commands into a simple sequence – a content to which programming in K–3 is commonly limited (Zhang & Nouri, 2019). In the following, we present research focusing on aspects of what it means to understand the meaning of programming, as well as to learn to program in terms of ordering commands into a simple sequence.

The concepts input and output are vital parts of the main idea of programming. Burke and Kafai (2012, p. 438) describe that "coding is an attempt to articulate a precise input to facilitate a particular output". In other words, learning to program involves learning that programs require input of data, that this data is processed by the "processing system", and that "signals or data" sent from the system are the result (output). A common analogy for computer programming especially in primary school is that of "the computer as a chef and the program as a recipe". This is exemplified in the classic jam sandwich activity, where the teacher takes the role of a robot, and the students are to instruct the robot to make a jam sandwich. The purpose of the activity is to illustrate how things can go wrong (output) when the instructions are not precise enough. When introducing computer science to middle school students, Sivilotti and Laugel (2008) used the same analogy but with a different focus. The point made was that a "chef is a general-purpose processor, capable of transforming ingredients (input) into a final dish (output), while the recipe is the sequence of instructions for how to carry out this transformation" (2008, p. 292).

For novices, understanding the meaning of programming assumes, to some degree, an understanding of what a computational device may, or may not, “understand” in relation to a human. In other words, this understanding may be formulated as being able to discern the differences between the artificial and the human mind. Understanding the artificial mind means understanding that the robot has no will; its actions depend on direct operation or programming (Spektor-Precel & Mioduser, 2015b, 2015a). Pea (1986) explicates a central aspect of what it means for a computer to read a program when emphasizing that “programming is a formal system that interprets each part of a program (instructions to it) in terms of rules that are mechanistic” (1986, p. 26) and adds that “while people are intelligent interpreters of conversations, computer programming languages are not” (p. 26). Pea explains how this feature violates the rules of human conversation: for example, a computer “cannot infer what a speaker means if she is not absolutely explicit, whereas a listener in a human-human conversation can query the speaker for clarification” (1986, p. 26).

To be able to identify and create correctly sequenced programs to accomplish a specific goal or action, one needs to understand the temporal logic of sequential instructions. To have this sequencing ability may be described as having the ability to place objects or actions in a correct order: first – then – last (Zelazo et al., 1997). In their literature review of computer science education research focusing on primary school, Rich et al. (2017) identified three consensus goals concerning the aspect of order: 1) Different sets of instructions can produce the same outcome, 2) The order in which instructions are carried out can affect the outcome, and 3) Computers have a default order of execution, so order matters in programming. Learning to order commands into a sequenced program therefore means learning to discern and understand these three aspects. Notably, they directly (1 and 2) and indirectly (3) involve the concept of output. These goals can be considered as tentative critical aspects when planning for UP instruction focusing on the sequencing of commands.

2.2 Computational thinking and unplugged activities

Learning programming is expected to help students develop computational thinking. Wing (2006) describes computational thinking as a set of fundamental skills which help us understand and solve problems in the same ways as computer scientists do. A few years later, Wing (2011) defines computational thinking as the thought processes involved in formulating problems and their solutions, where the latter are expressed in such a way, that is algorithmically, that a computer or human can execute it effectively. The centrality of representing solutions in ways that can also be understood by humans, i.e., including humans as information processing agents, mirrors Wing’s emphasis on the importance of including computational thinking in early childhood education. Manches and Plowman (2017, p. 6) describe how Wing’s broadened notion of computational thinking “allow(s) for a graduation from general communication skills to the specific practice of coding” and allows for activities aligning with early years pedagogy – for example, giving a human "robot" instructions to make a jam sandwich. In other words, computational thinking does not require a machine. Everyday activities, such as delivering and collecting things or making breakfast, involve computational thinking. Bell and Lodi (2019) point out that matching UP activities with computational thinking ideas is useful since they serve a similar purpose: UP activities are rooted in the intention "to help children to understand what a computer scientist does". Importantly, Wing (2006, 2011) refers to computational thinking as "thinking like a computer scientist" (2011, p. 345). She emphasizes CT as a "fundamental skill" on the same level as reading, writing, and arithmetic. Moreover, she stresses that it is not only about programming, but about ways of organizing thought that can be developed from an early age. Definitions of computational thinking varies throughout the literature. However, the components common among researchers are: decomposition (break a problem down into doable units), abstraction (modeling the relevant aspects of a problem), algorithms (developing step-by-step instructions to solve a problem), and debugging (finding and fixing errors) (Shute et al., 2017).

2.3 Aspects of instructional approaches to programming in primary school – prior research

Research concerning the development of UP learning activities emphasizes the importance of analogy and kinaesthetic activity (Curzon et al., 2009) including implementation with concrete hands-on practices that enable children to manipulate codes (Lee & Junoh, 2019). Sung et al. (2017) investigated the way in which UP activities affected, among others, computational skills. They found that enacting full-body movements led to better performance on mathematics tests and programming with ScratchJr, than activities requiring a lower level of embodiment (hand movements along a number line on a piece of paper). The authors concluded (2017, p. 449) "that a greater degree of bodily engagement supports the perceptual experiences of learners by providing concrete experiences".

For both UP and plugged-in activities, providing a real-world context is stressed since it is believed to provide a meaningful learning context (Campbell & Walsh, 2017; Grover & Pea, 2013; Lee & Junoh, 2019). Studies show that younger children find it easier to understand sequences of which they have their own experience (Fivush & Mandler, 1985), and that sequencing capabilities are higher when the task is placed within a meaningful context, such as a narrative (Brown & Murphy, 1975). From their literature review, Zhang and Nouri (2019) conclude that students in the identified studies comprehended sequences relatively easily since they could use analogies from daily life to conceptualize the concept as a list or series of instructions. Contextualization was also highly emphasized by respondents in Sentance and Csizmadia's (2015) survey of UK teachers' perspectives on strategies for teaching computing, in terms of relating it to other subjects as well as to real-life activities (e.g., making a cup of tea).

3 Theoretical framework

In this article, where we are interested in what learning is made possible in a UP lesson, we have, as stated above, chosen variation theory as a theoretical framework (Kullberg et al., 2024; Marton, 2014; Runesson, 2017). The rationale for this choice is based on the fact that variation theory allows for a more detailed content analysis.

3.1 Variation theory, object of learning, and critical aspects

Variation theory is a theory for both designing and analysing teaching (Kullberg et al., 2024; Marton, 2014; Runesson, 2017). As stated in the introduction, the concepts of object of learning and critical aspects are central to variation theory. The first concept, object of learning, is used to define the precise knowledge of a particular content that students are required to develop. An object of learning thus consists of a specific content and the students' expected capability to use this content in specific ways. Defining the object of learning requires an analysis of both the subject-specific knowledge and students' prior experience of it.

From the perspective of variation theory, the way one experiences something depends on what aspects one discerns, and to what extent one can experience several aspects simultaneously (Kullberg et al., 2024; Marton & Fung-Lo, 2007). This means that when students in the same classroom experience a particular content or phenomenon– the object of learning – in different ways, this is because they discern different aspects of it. The aspects of the learning object that students need to discern in order to qualify their knowing are conceptualized as critical aspects. In other words, learning is "becoming able to discern critical features of an object of learning. The object of learning concerns a capability or understanding of something, for example a particular content taught in school" (Kullberg, 2010, p. 33).

The object of learning may be studied from different perspectives. The intended object of learning refers to what the students are expected to learn – the teacher's aim for the lesson. The enacted object of learning refers to what students had the possibility to experience during the lesson (what content is made visible in the classroom). The lived object of learning refers to students' knowledge after the lesson. In this article, we focus mainly on the enacted object of learning to be able to say something about what learning was made possible during the analyzed UP lesson.

4 Method

To be able to explore what aspects were made available, and which of them seem to be critical, during the UP lesson, the study from which data is drawn was conducted in a qualitative research tradition. The study therefore has a descriptive character including detailed descriptions of who (teacher and students) does what, with which tools, and why.

4.1 Data collection and analysis

Data were collected from observations of one technology lesson on programming in grade 1 (6–7-year-olds). To begin with, the teacher divided the class in two. One half of the class (1) stayed in the classroom, and the other half (2) was taught by another teacher. When the lesson with half-class 1 was finished, the lesson was repeated with half-class 2. The observations of half-class 1 and 2 were documented using three video cameras and three separate microphones. During teacher-led reviews, one camera was placed at the far back of the classroom, and the other two cameras and microphones along the right- and left-hand walls, respectively. The researchers stood behind, or moved between, the three video cameras to follow the teachers' actions, including the teacher-student interactions. When the students worked together in groups of two, the recordings were focused on three student groups in each half-class (a total of six groups) to capture the students' actions. These focus groups were chosen randomly "in the moment". It later turned out that the voice recordings of three of the six student groups failed. This meant that the analysis of the empirical material was restricted to three groups of students in total.

The communication in the classroom and within each of the three student groups was transcribed from video and voice recordings. The data preparation of the transcribed video recordings took place in three phases. Phase 1: We divided the recordings into four parts corresponding to the four classroom activities that made up the lesson. Phase 2: For each part, we firstly identified events in which some reference to computing and technology content was noticeable and then segmented these into episodes – independent units consisting of pieces of dialogue that shared the same focus (Gee & Green, 1998). The boundaries were determined by shifts in the dialogue and/or activities. Phase 3: We transcribed each of the episodes including notes about the students' gestures. The episodes were translated to English by the authors. We wish to emphasize that nuances of the original may have been lost in translation.

The analysis of the transcribed material was conducted from the perspective of variation theory, focusing on what aspects were constituted in the classroom and which of these can be regarded as critical. To identify what aspects could be seen as critical, the analysis focused on the different ways in which the students expressed their understanding of the object of learning. More precisely, to recognize what emerges as a critical aspect in an ongoing lesson, attention needs to be paid to situations where students, for example, give explanations that contain a contradiction. This gives an indication that there is an aspect that the student has not yet discerned.

4.2 The lesson

The observed lesson was planned by the teacher using a selection of activities from a book ("How to Code" [in Swedish "Lär dig koda] by Wainewright, 2016) of the teacher's choice. The lesson thus consisted of four activities, see Table 1. (Activities 1–4). The focus of the present article is Activities 2–4 since the analysis of Activity 1[1] is presented in Berg and Axell (2023).

Table 1. Overview of the four activities during the lesson.

Activity		
The relation between robot and human	See footnote.	
The jam sandwich	The teacher shows a video clip from a classroom in which the students are tasked with programming their teacher (verbal instructions) to make a sandwich with strawberry jam.	
Robot breakfast	The students carry out the task Robot breakfast which is to instruct an imaginary robot to "make breakfast." They were given a worksheet with five written commands and were instructed to place these in the right order.	
Teacher-led follow-up of the task Robot breakfast	During the review, some of the student groups were asked to present their solutions, and the teacher recorded the sequences on the whiteboard.	

4.3 Ethical considerations

Ethical considerations were followed according to the recommendations of the Swedish Research Council (2017, 2024). Written information was sent to the students' legal guardians including a description of the study, a letter of consent regarding video and audio recording of the students in class, and information that they had the opportunity to withdraw their consent at any time. Two legal guardians did not express consent. Thus, when documenting the students' work in the classroom, we chose camera angles that would not include these two students. The names of the students are pseudonymized in order to de-identify all participants.

All data material including consent forms was handled and stored according to current rules at Linköping University.

---

[1] During Activity 1, the teacher staged a sequence of events well in advance of the start of the lesson by turning off the light and opening a window in the classroom. Concurrently, it was cold and dark in the classroom when it was time for the students to enter. The teacher began the lesson by stating that it was "somewhat dark and cold" in the classroom. As a response to this, one student turned on the light, and another closed the window. Then followed a conversation between the teacher and the students focusing on the difference between the human's and the robot's mind, and the meaning of the concept of programming. Critical aspects that were discernable during Activity 1 and of interest for the present study included a) humans can understand what someone else means without them explicitly expressing it, while robots lack emotions and therefore cannot do this, and b) robots, unlike humans, must be "told" what to do.

5 Results

Overall, based on the analysis, we can discern that to develop the ability to understand the idea of sequencing of commands, students need to discern a number of quite detailed aspects. The analysis of students' actions indicates that some of these are understood by students while others appear to be more difficult for students to discern – these aspects thus appear to be critical. The Results section is divided into three subsections, following the chronological order of Activities 2–4.

5.1 Activity 2: The Jam Sandwich – A robot needs clear instructions

The students watch the video The Jam Sandwich and the activity concludes with a short whole-class reflection on how to instruct a robot. One student states that the robot "didn't succeed", the reason being that "the robot did exactly as they [the students in the video] said and if they said something wrong or crazy, the robot did it anyway". In other words, the student expressed a central aspect of the object of learning. The teacher links this aspect to the concept of programming when she replies that "that can happen if you haven't programmed [...] the robot to do the right things". Notably, what it meant to say something "wrong", "crazy" or "right" to the robot was not further elaborated on. The teacher emphasized the importance of clear instructions: "If you are a robot, you need to have clear instructions to know exactly how to do it, because the robot does not think for himself, he just listened to the instructions." The aspects (hereafter denoted by "A") that are constituted during this episode are summarized in Table 2 and may be described as "A robot does exactly what it's instructed to do even though the instructions may be wrong or crazy" (A1) and "A robot needs clear instructions since it can't think for itself" (A2). Based on the teacher's utterances, the purpose of showing The Jam Sandwich can be described as demonstrating a critical aspect of programming in practice, that is, the robot's artificial (non-human) mind and hence the importance of clear instructions.

Table 2. Summary of tentative critical aspects identified in Activity 2: The Jam Sandwich

Category of content	**Activity 2: The Jam Sandwich** Aspects of the object of learning that are implicitly or explicitly expressed in the classroom
The mind of the robot	A1: A robot does exactly what it's instructed to do even when the instructions may be wrong or crazy. A2: A robot needs clear instructions since it can't think for itself.

5.2 Activity 3: Making breakfast – Aspects of the object of learning sequencing of commands

During Activity 3, the students, working in pairs, carry out the task Robot Breakfast in which they are supposed to instruct an imaginary robot to "make breakfast". In the following, we firstly describe the teachers' introduction of the task and secondly, the students' actions within three of the student groups.

5.2.1 Introduction of the task Robot Breakfast

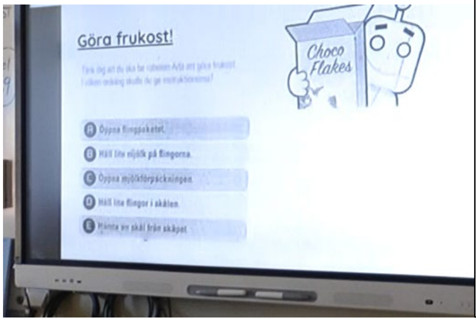
The activity begins with the teacher projecting the worksheet for the assignment on the smartboard. Under the heading Making Breakfast, there is an instruction for the students: "Imagine that you are going to ask the robot to make breakfast. In what order would you give the instructions?" This is followed by five commands (see Fig. 2). Performing the task Robot Breakfast therefore means correctly ordering these five commands. The overall purpose of the task, as we read it, is to give students the opportunity to experience sequencing with a focus on the aspect of order. As mentioned, we interpret the intended object of learning as the capability to understand the idea of sequencing of commands.

The teacher then tells the class that they "should think about in what way the robot should make breakfast so that it turns out well and not as crazy as for him in the video here". Next, she asks students to read aloud the five instructions, and then she comments on their order: "This is not in a good order, perhaps, if you were to make breakfast," and some students answer with a "no". The teacher then instructs the students on how to complete the assignment:

> Teacher: You should think about which order would be the best to give these instructions to the robot. There are several solutions that work. I'm sure there are a lot of solutions that don't work. But try to find several solutions that are possible.

In conclusion, the teacher emphasizes three tentative critical aspects (see Table 3), the first being that the order of commands matters. She implicitly expresses that there is a relationship between the quality of the order and the quality of what is accomplished – a good order leads to a good (i.e., not a badly-made) breakfast (A3). The teacher also emphasizes that there are several solutions that work and several that don't work, i.e., different sequences of commands can produce the same outcome (A4). The meaning of "a solution that works", or not, is however not explicated. Finally, as the teacher states that the given order of commands is "not good", and that the students are to think about "the best" order, she implicitly expresses the critical aspect that programming may – as in this case – be about finding and fixing the lack of good order in a sequence of commands (A9). This critical aspect concerns what may be described as the perspective of the programmer (see Table 3).

Figure 1. The instructions for the Robot Breakfast task are displayed on the Smartboard (left). Students arrange the five commands in a specific sequence by writing a number (1-5) next to each command (right).



A Open the package of cereal
B Pour some milk on the cereal
C Open the milk carton
D Pour some cereal into the bowl
E Pick up a bowl from the cupboard

5.2.2 Martin and Leo: A sequence of commands must be based on temporal logic

Below follows a description of how the solving of the task evolved in the group consisting of Martin and Leo.

Martin quickly takes the lead; he points to the command "Pick up a bowl from the cupboard" and says, "That one first." Leo asks, "What is [that]?" and Martin reads the instruction. He then quickly chooses what the next instruction should be: "Open the cereal box. A [number] two there", and identifies what he says must "absolutely be [command number] three" – "Pour some cereal into the bowl". Leo reacts strongly to this:

> Leo: No! You should pour milk first.
>
> Martin: No.
>
> Leo: You pour <u>milk</u> first.
>
> Martin: No.
>
> Leo: Yes Martin, milk instead, Martin.
>
> Martin (points at the paper and reads): There, "Pour some milk on the cereal".
>
> Leo (irritated voice): OK then. Great fun to work with you [writes a "3" next to the command "Pour some cereal into the bowl" and then turns away from Martin].

Leo's objection is seemingly about the order of commands chosen by Martin, which conflicts with Leo's everyday experience and personal preference – "You pour milk [into the bowl] first [and then the cereal]" – and not about a belief that the command "Pour cereal into the bowl" leads to a sequence that lacks temporal logic. However, Martin tries to draw Leo's attention to the meaning of command B, "Pour some milk on the cereal". Leo does give in but clearly communicates that he doesn't agree with Martin. Martin then proceeds to point out the order of the two remaining commands (4, “Open the milk carton” and 5, “Pour some milk on the cereal”). At the same time, Martin argues for his choice: “You can’t pour the milk in... on the cereal before you... before the milk carton has been opened.” Leo has obviously been able to discern that the command “Pour some milk on the cereal” not only requires that the cereal is already in the bowl, but also that the milk carton has been opened. Leo takes notes, but then demonstratively turns away from Martin. To conclude, Martin has firstly discerned the overall aspects “A robot does exactly what it’s instructed to do” (A1) and “A sequence of commands must be based on temporal logic” (A5, Table 3). The former aspect is a prerequisite for the latter, that is, when sequencing commands based on temporal logic, the programmer needs to consider every word in the commands. Finally, Martin has also discerned the aspect that prerequisites shape the temporal logic of the instruction sequence. This aspect may be described as understanding that the mutual order of two commands may be fixed if the first command shapes the conditions necessary for the next command to be executed (A6, Table 3).

When the teacher passes Martin and Leo, she asks them if “there is any other solution that also works”. Martin and Leo lean over the paper, and quickly Martin says, “That one” and points to one of the commands (it cannot be seen which one). Leo, however, objects and says that Martin is “just saying the wrong thing” and that it is Leo’s turn to “decide”. Martin then hands over the choice to Leo.

> Martin: Now you can decide on one [command]. [Martin looks at the worksheet and seems to be thinking.] But this one also fits [points at some of the commands, unclear which].
>
> Leo: Where's the bowl then? The bowl is the first [writes a “1” next to the command “Pick up a bowl from the cupboard”]. [...] then... number two is open the milk [writes “2” next to the command “Open the milk carton”.]
>
> Martin: And by the way, you can open...

Leo keeps “Pick up a bowl from the cupboard” as his first command but then selects “Open the milk carton”. Martin then suggests another command (which is probably “Open the package of cereal”). However, Leo wants to have “Pour some milk on the cereal” as his next command:

> Leo: And then pour some milk into, pour some milk.
>
> Martin: No, it’s “Pour some cereal into the bowl” and then it’s “Pour some milk on the cereal”.
>
> Leo: No, but you don’t have to do that Martin. But then pour some milk, or like this, pour some milk, where does it say? [reads the worksheet].
>
> Martin: It doesn’t say anywhere.
> Leo: Pour some milk on the cereal.

Martin protests and emphasizes, once again, that the milk should be poured on the cereal. However, Leo believes that "you don't have to do that", which can perhaps be interpreted as meaning that, when you prepare your breakfast, you can do what you want – cereal or milk first. Martin helps Leo find "Pour some milk on the cereal" and Leo immediately writes a "3" in front of the command. Next, Leo selects "Open the cereal package" and "Pour some cereal into the bowl" as the fourth and fifth commands, respectively. Martin then points at "Pour some cereal into the bowl" while saying, "But then you have to have the cereal before," to which Leo replies, "But you can do that anyway." However, Martin has a hard time accepting Leo's choice and emphasizes that "that's what it says". Leo stands his ground but cannot formulate any other argument than "No but Martin, still, it can be that one [command "Pour some milk…"] first." Then he writes 4 in front of "Open the package of cereal" and 5 in front of "Pour some cereal into the bowl".

To conclude, Martin's reasoning rests on the overall aspect: the commands must be ordered based on temporal logic (A5, Table 3) and the specific case of A6: the order of two commands matters when one command shapes the prior conditions necessary for the other command to be executed. Martin has also discerned the aspect that the content of a command is not negotiable – "A robot does exactly what it's instructed to do" (A1), or in other words a robot will "read" the commands literally. As a part of this, he has also discerned that the robot can't think for itself (A2). Leo, however, has obvious difficulties understanding what frames the task. It thus appears as if the aspects A1-2 and A5-6 are critical for him.

5.2.3 Tomas and Tim: The order of instructions doesn't always matter

Below follows a description of how the solving of the task evolved in the group consisting of Tomas and Tim.

The boys agree on "Open the package of cereal" as the first command and "Open the milk carton" as the second. Tim then reads aloud: "Pick up a bowl from the cupboard" and "Pour some cereal into the bowl" and writes a "3" and a "4" next to the commands, respectively. Finally, Tim writes the number 5 next to "Pour some milk on the cereal". Tomas is silent but nods a little when Tim is finished. In summary, Tim's way of acting implies that he has discerned A5 - the ordering of the five commands should be based on temporal logic – and that the task does not challenge him. As for Tomas, his silence implies insecurity about the meaning of the task. This indicates that A5 is critical for Tomas.

When the teacher then asks if they "can find any more solutions", Tomas shakes his head. The teacher asks them to "think about it", whereupon Tim says, "Yes there is one more that I know" and points to one of the commands (unclear which one). In the meantime, Tomas comments on Tim's statement: "Well, it's the same just that you pour the cereal first." Here it appears that Tomas' interpretation of the sequence they have just arranged is that first the milk is poured into the bowl, and then the cereal, and that by rearranging these commands you get a new solution. In other words, he expresses that the order in which the milk and cereal are to be poured into the bowl is optional. This implies that Tomas, just like Leo above, has not been able to discern the meaning of A6 – "The order of two commands matters when…" – and thus what the robot will ultimately achieve. This implies that this aspect is critical for Tomas.

Tim disagrees with Tomas about his suggestion and says "No, you do like this" while writing "1" next to "Open the milk carton". Tomas reacts with a "Huh?" but Tim doesn't respond and continues to write a "2" in front of "Open the package of cereal". Compared to their first solution, Tim thus reverses the order in which the milk and cereal are to be opened. Tim then proceeds to arrange the remaining three commands in the same order as in the first solution. Tomas looks away while Tim takes notes, but when Tim explains that it is "done", Tomas turns to the worksheet and questions this:

> Tomas: I can… is that all? But number 5? [points at the worksheet]
>
> Tim: Yes, that one [writes "5" next to "Pour some milk on the cereal"]
>
> Tomas: What is number 5? What happens?
>
> Tim: Pour some milk on the cereal.

Tomas' question about the fifth command implies that he has not discerned that "Pour some milk on the cereal" must always come last. Hence, it further confirms what is implied above – Tomas has difficulty discerning the prerequisite embedded in the command, and the sequencing of the five commands as accomplishing a specific outcome. The fact that Tim marks only instructions 1–4 can be interpreted to mean that for him, this is obvious.

To conclude, Tim discerns that the sequence of "Open the package of cereal" and "Open the milk carton" doesn't affect the outcome. He thus expresses another aspect: the sequence of two commands doesn't matter if the prior condition for one command is independent of the action of the other (A7, Table 3). In other words, the order of commands doesn't always matter. As for Tomas, it seems as if he hasn't been able to discern what Tim has. Thus, A7 is also critical for Tomas.

5.2.4 Rosa and Linn: When a sequence of commands is not based on temporal logic, something undesired is accomplished

Below follows a description of how the solving of the task evolved in the group consisting of Rosa and Linn.

Rosa and Linn have chosen "Pick up a bowl from the cupboard" as their first command. Rosa reads out loud and suggests "Pour some cereal in the bowl" as the next command. Linn first agrees but then she changes her mind:

> Linn: No, no, open the package of cereal.
>
> Rosa: Open, open.

> Linn: Here [points at the command "Open the package of cereal" on the worksheet].
>
> Teacher [when passing by Rosa's and Linn's bench]: Is it important to do that ["Open the package of cereal"] before [one pours the cereal]?
>
> Linn: Yes.
>
> Teacher: Why?
>
> Linn: Because you, otherwise you can, otherwise you just pour [the cereal] inside the package of cereal.
>
> Teacher: Yes, exactly.
>
> Rosa: My box is down here [turns around and points].
>
> Linn: [inaudible] this now.
>
> Rosa: [inaudible]
>
> Linn: Yes. Write now.

With her questions, the teacher makes Linn formulate an argument for her choice: "otherwise you just pour inside the package of cereal". Obviously, Linn has discerned that temporal logic demands the specific sequence of the two commands, and that overlooking this results in something undesired. Rosa's way of directing the attention away from the task may be interpreted to mean that she does not understand the meaning of what Linn is explaining to the teacher.

Linn then turns her attention to the choice of command number three, whereupon Rosa reads aloud: "Pour some milk on the cereal." Linn, however, protests:

> Linn: No, no [points at the worksheet and reads aloud] pour, pour the cereal in the bowl, pour the cereal in the bowl. OK. There, number three.
>
> Rosa: Pour some cereal, some cereal in the bowl.
>
> Linn: In the bowl.
>
> Rosa: Then this should be number three [writes "3" next to the command].
>
> Linn: Yes.

Linn protests since she has obviously discerned the embedded prerequisite in "Pour some milk on the cereal". Rosa accepts Linn's "no" but doesn't explicitly express that she agrees. Linn then points to the instructions "Pour some milk on the cereal" and "Open the milk carton" and says "four" and "five" respectively, and then suddenly stops:

> Linn: Wait what does this say, open milk, no! This [points at “Open the milk carton”] should be four, this [points at “Pour some milk on the cereal”] should be five.
>
> Rosa: [inaudible] erase.

Again, when she reads the instructions one more time, Linn discerns that the temporal logic fails; the milk carton must be opened before the milk may be poured. Rosa acts as in the situations above – she accepts without confirming.

In conclusion, Linn, like Martin and Tim, approaches the task from the perspective of temporal logic. During her communication with the teacher, Linn also expresses the consequences of missing out temporal logic: one pours the cereal within the package. In other words, Linn has discerned the aspect “When a sequence of commands is not based on temporal logic, something undesired is accomplished” (A8, Table 3). This, in turn, rests on an understanding about how robots “read” instructions. Thus, Linn has also discerned the aspect “A robot does exactly what it’s instructed to do” (A1). Rosa, however, has seeming difficulty discerning these aspects and they are potentially critical for her.

The aspects and critical aspects identified in Activity 3 are summarized in Table 3 below. These aspects may be divided into three different categories of the object of learning. The first category includes an aspect related to the mind of the robot while the second consists of six “rules” of sequencing. The third category includes one aspect that concerns the work of the programmer. We choose to call this category the perspective of the programmer.

Table 3. Summary of aspects and critical aspects identified in Activity 3: Robot Breakfast.

Category of content	**Aspects (A) of the object of learning that are implicitly or explicitly expressed during Activity 3**
The mind of the robot	A1: A robot does exactly what it’s instructed to do even when the instructions may be wrong or crazy.
Sequencing logic	A3: The order of commands matters; a good order leads to a good outcome (Introduction). A4: Different sequences can produce the same outcome (Introduction). A5: A sequence of commands must be ordered based on temporal logic. A6: The order of two commands matters when one command shapes the prior conditions necessary for the other command to be executed. A7: The order of two commands doesn’t matter if the prior condition for one command is independent of the action of the other. A8: When a sequence of commands is not based on temporal logic, something undesired is accomplished.
The perspective of the programmer	A9: Programming may – as in this case – be about finding and fixing the lack of good order in a sequence of commands. (Introduction).

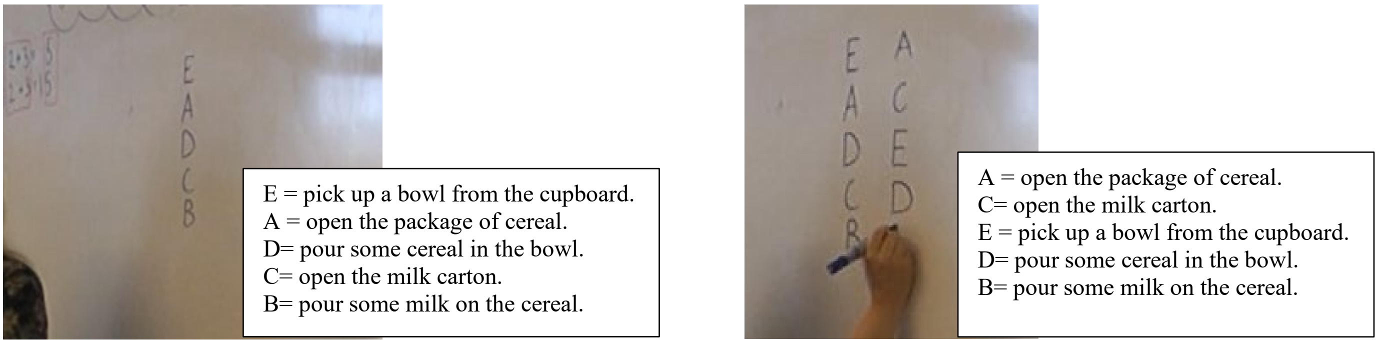
5.3 Activity 4: Teacher-led review on the task Robot Breakfast

During Activity 4, the teacher leads a follow-up on the task Robot Breakfast. During the follow-up, some of the student groups were asked to present their solutions, and the teacher recorded the sequences on the whiteboard in the form of columns of letters, each letter denoting a given instruction (see Fig. 2). The content that took form in each of the two half-classes during the follow-up was similar. We therefore chose to have the episodes framed by a chronological description of the teaching in half-class 1, but supplement with two episodes from half-class 2 which in their content differs little from the corresponding episodes in half-class 1.

5.3.1 Aspect 10: The meaning of a solution that works is that nothing crazy is accomplished

The teacher begins the review by asking Maria and Melwin (not video/audio recorded during their work on the task) to report their solution to the task. As the students read their sequence aloud, the teacher writes down the letter of each command in a column on the whiteboard (see Fig. 2a).

Figure 2. Figure 2a (left). The teacher has noted Maria and Melwin´s solution on the board. Figure 2b (right). The teacher is noting Julia and Jan's solution (right-hand row) next to Maria and Melwin's.



Then she asks Julia and Jan (not video/audio recorded) to present their solution, encouraging them by saying that "it doesn't matter if it goes crazy". When their solution has been listed on the board by the teacher (see Fig. 2b), she asks the class if the "solution works" and some students answer yes. The teacher then asks a follow-up question which implicitly expresses the meaning of "works" as "nothing crazy happens". Presumably, she is referring to the jam sandwich video (Activity 2) in which several crazy things happened and therefore does not elaborate on the meaning of crazy. To conclude, the teacher implicitly highlights an aspect that may be formulated as: "The meaning of a solution that works is that nothing crazy is accomplished" (A10, Table 4). As the reverse also applies, A10 may equally be formulated as "The meaning of a solution that doesn't work is that something crazy happens". However, for the students to be able to discern the meaning of a crazy or a non-crazy outcome, and thus to experience the object of learning in a more complex way, they also need to discern and attend to the desired output of the five commands.

5.3.2 Aspect 8: When a sequence of commands is not based on temporal logic, something undesired is accomplished

The next solution presented is William's (William worked alone and was not video/audio recorded during his work on the task). The teacher takes notes on the whiteboard and then reads aloud the sequence: "Open the cereal box > Pick up a bowl > Pour some milk on the cereal > Open the milk carton > Pour cereal into the bowl." The teacher then asks the class the same question as before, "Does it work?" From the students, you can hear scattered calls of "yes" as well as "no", and it is obvious that not all of them have discerned that the chosen sequence does not work. When the teacher then asks for an explanation, Kim says, "First you said pour some cereal and then you said open the cereal." In other words, Kim implicitly highlights that the sequence lacks temporal logic (A5) and rests her reasoning on A6: "The order of two commands matters when one command shapes the prior conditions necessary for the other command to be executed." When the teacher then asks the class "What do you have to do?" Karl replies "Open [the package] first." The teacher confirms this and, in line with Martin in his discussion with Leo, uses and emphasizes the word "before": "You must open the package before you pour it." She thus implicitly highlights A6. From the context follows implicitly that if this is not fulfilled (temporal logic), the solution does not work, that is, an aspect corresponding to A8. However, the outcome of instructing the robot with the non-working solution is not elaborated on. Thus, the students are not enabled to discern in what way a lack of temporal logic amounts to a non-working solution (A8) or what a non-working solution as such really means. In other words, the constituted content is devoid of an aspect that may be formulated as "The meaning of programming is to accomplish a desired and specific output".

5.3.3 Aspect 11: The meaning of a solution that doesn't work is that something undesired is accomplished

Also in half-class 2, two solutions that work are presented. The teacher then turns to Martin and Leo and asks if they have any other solutions than the two presented on the board. Leo answers that they don't, adding that "it went wrong". Presumably, Leo came to this conclusion when he heard another group's solution, because he then erased his notes on the worksheet. The teacher asks if they should look at the solution anyway, and Leo reads the sequence out loud: "Pick up the bowl > Open the milk carton > Pour some milk on the cereal > Open the cereal box > Pour some cereal into the bowl."

The teacher notes the sequence on the board and then asks why they erased the solution. Leo seems to have difficulty formulating an answer while Martin describes the problem:

> Leo: Eeeh...
>
> Martin: You can't pour milk on cereal when you don't have any cereal [in the bowl]. [At the same time, Leo is quiet and looks out the window]
>
> Teacher: No. Where does the milk end up in that case.
>
> Leo: In the bowl.

Martin's argument has a different focus compared to when he was working with Leo but also compared to the teacher in the preceding episode. Instead of describing what ought to come first, he now focuses on a situation where the non-working sequence has been executed and describes the problem that arises: there is no cereal in the bowl for the milk to be poured on. Hence, his argument can be explained by A6 and he links the violation of this "law" to a non-working solution. This focus on the outcome of the non-working solution invites the teacher to further unpack what Martin started: "Where does the milk end up in that case?" Now, Leo turns to the teacher and says, "In the bowl." This shows that, after all, he has not discerned what is wrong with his solution. The teacher responds to Leo's answer in the negative and then two other students come up with another answer:

> Nelly: On the package.
>
> Saga: On the package.
>
> Teacher: On the package, yes.
>
> Nelly: If they haven't opened the package of cereal.
>
> Teacher: It was lucky that you reprogrammed your robot otherwise it would have been milk on the cereal package yes.

Unlike Leo, both Nelly and Saga suggest that the milk will end up wherever the cereal is. In other words, they have discerned the critical aspects linked to the mind of the robot (A1-2). Their answer, "On the package," implies an idea that the robot "knows" that there is cereal inside the package and therefore pours the milk on the package. A scenario in which the program halts due to an absence of cereal is obviously not what comes to the students' minds. Presumably the jam sandwich video influences their reasoning: the robot in the video did crazy things in line with pouring milk on a package of cereal, but it never happened that it stopped doing something.

The answer by Nelly and Saga, "On the package," implicitly describes how one knows that the sequence doesn't work: milk being poured on the package is in line with the tomfoolery in the video. When the teacher confirms the students, she also signals that milk on the package is something undesired – "it was lucky that you reprogrammed your robot". Hence, she implicitly expresses A11: "The meaning of a solution that doesn't work is that something undesired is accomplished" (Table 4). However, what is not simultaneously in focus is the desired outcome.

5.3.4 Aspect 12: The order of two commands in a solution that works can change and the result be yet another solution that works

A third aspect, A12, is also constituted during Activity 4. When Julia and Jan's solution has been listed on the whiteboard and assessed as a working solution, the teacher points at the solution (Fig. 2b) and directs the students' attention towards the fact that "there is another [solution in addition to Maria and Melwin's]". Melwin comments on this by saying "There were some that changed places" – a comment that the teacher endorses. Melwin has obviously recognized a pattern: the order of instructions A and E, and then D and E, differs between the two solutions (see Fig. 2b). The teacher confirms Melwin but doesn't elaborate on his finding. Thus, what is implicitly expressed by Melwin and the teacher, is an aspect of the object of learning that may be formulated as: the order of two commands in a solution that works can change and the result be yet another solution that works (A12, Table 4). How this is possible is, however, not highlighted. Thus, A12 is devoid of explanatory information corresponding to A7 – "The order of two commands doesn't matter if the prior condition for one command is independent of the action of the other".

The three identified potential critical aspects in Activity 4 are summarized in Table 4 below. These aspects may be divided into two categories of content: The output and Sequencing logic.

Table 4. Summary of tentative critical aspects identified in Activity 4: Teacher-led review of Robot Breakfast.

Focus of content	Aspects of the OL that are implicitly expressed in the classroom
The output	A10: The meaning of "a solution that works" is that nothing crazy is accomplished A11: The meaning of "a solution that doesn't work" is that something undesired is accomplished A8: When a sequence of commands is not based on temporal logic, something undesired is accomplished.
Sequencing logic	A12: The order of two commands in a solution that works can change and the result be yet another solution that works.

6 Discussion

In the following, we turn to discuss some of the findings presented above. We depart from the questions: What aspects of the object of learning can be distinguished in the student-teacher actions during UP activities and which of these aspects can be identified as critical? How can the UP activities be enriched to allow students to experience the sequencing of commands in a more qualified way?

We have identified several aspects that are constituted during different activities related to the task Robot Breakfast. Based on the students' different ways of expressing their understanding, we may conclude that five of these aspects appear as critical (A1–2 and A5–7):

A1: A robot does exactly what it is instructed to do, even when the instructions may be wrong or illogical.
A2: A robot needs clear instructions since it cannot think for itself.
A5: A sequence of commands must be ordered based on temporal logic.
A6: The order of two commands matters when one command shapes the prior conditions necessary for the other command to be executed.
A7: The order of two commands does not matter if the prior conditions for one command are independent of the action of the other.

The results of this study also illustrate that the aspects identified during Activity 4 – A10-12 – are not detailed enough; they do not fully capture the complexity of understanding what constitutes a correct or incorrect solution. Consequently, these aspects may not sufficiently help the students to discern the meaning of a solution that works and doesn't work, and why two different solutions may work equally well.

In the following, we discuss the aspects A1–2, A5–7 and A10–12, as well as suggest how Robot Breakfast may be developed to further support students' discernment of critical aspects and thereby experience the object of learning in a more qualified way.

6.1 Critical aspects 1–2: The mind of the robot

The analysis of the students' interactions illustrates that Martin, Tim and Linn understand and act according to the aspects related to the "mind" of the robot (A1-2). However, these aspects are critical for Leo, and presumably also for Tomas and Rosa. Leo's reasoning indicates that he assumes the robot can interpret the instruction "Pour some milk on the cereal" even when placed before "Pour some cereal in the bowl". His approach mirrors conversational logic rather than computational logic, treating the robot as an intelligent interpreter that can "read between the lines" (Pea, 1986).

Importantly, while Activities 1 and 2 addressed the concept of the robot's "mind", there was no explicit focus on the critical aspect that the robot follows instructions literally and cannot infer missing information. We propose refining A1 to: "The robot does exactly what it is programmed to do and cannot infer unstated information."

6.2 Critical aspect 5: A sequence of commands must be ordered based on temporal logic

The analysis of the students' work with Robot Breakfast shows that Martin, Linn and Tim understand the meaning of temporal logic and act according to the aspects A5–7. They recognize that programming involves finding and fixing errors in a given sequence of commands. However, Leo interprets the task within the frames of an everyday perspective – instruct the robot to make breakfast the way you want it – which leads him to overlook temporal logic. Considering the instructions by the teacher – "think about which order would be the best" – Leo's conclusion doesn't seem farfetched.

Tomas and Rosa did not propose alternative solutions, but their verbal expressions and body language suggested uncertainty or resignation. We argue that these difficulties are more likely due to the task's structure rather than an inability to apply temporal logic, something which we discuss further below.

6.3 Critical aspect 6: The order of two commands matters when one command shapes the prior conditions necessary for the other command to be executed

The critical aspect that stands out during the students' work with Robot Breakfast is A6. When Leo attempts to place "Pour some cereal in the bowl" last in the sequence, Martin responds: "No, it's 'Pour some cereal into the bowl' and then 'Pour some milk on the cereal'".

Despite Martins's insistence, Leo does not immediately recognize the flaw in his reasoning. The teacher later directs attention to the outcome of the incorrect sequence: "Where does the milk end up?" Leo's response—"In the bowl"—suggests that the task and verbal arguments alone were insufficient for him to fully discern A1–2 and A5–6. However, it may be that the correct answer to the question by Kim — "On the package"—helped Leo to discern why the faulty order is faulty. Kim's answer compares with Linn's inference during the group work: the package of cereal needs to be opened before one pours the cereal in the bowl "otherwise you just pour inside the package". Yet, for these kinds of inferences to support discernment of A5 and A6, an explicit focus on the "mechanism" of the faulty sequence is presumably needed: why does the milk end up on the package of cereal?

6.4 Improving UP activities to support discernment of critical aspects 5-7

The question is thus how an activity such as Robot Breakfast may be qualified so that more students are enabled to identify when the order of two commands matters (A5–7) and when it does not, as well as how the robot reads the commands. In the following, we suggest four strategies.

The first strategy focuses on how the task is staged. One of the key arguments for UP is that it allows students to physically enact code, which can support their development of programming skills (Aranda & Ferguson, 2018; Fadjo, 2012; Sung, Ahn and Black, 2017). However, Robot Breakfast does not naturally encourage bodily engagement with the commands; instead, students must mentally visualize the sequence and its logical structure. Following the arguments of Fadjo (2012), Sung, Ahn and Black (2017) and Arand and Ferguson (2018), letting the students act out their mental images of the sequence of commands through manipulation of perceptual objects – a bowl, a package of cereal, a bottle of milk – should support the imagining by providing concrete experience. We believe bodily engagement should enable the students to activate their everyday experience of making breakfast, and that the task should work as a tool to mediate A6 as well as A7 ("The order of two commands doesn't matter…").

The second strategy draws on variation theory (Kullberg, Ingerman and Marton, 2024). When applying a variation theory perspective, dimensions of variations are used to help students discern a critical aspect. Regarding the task Robot Breakfast, simultaneously contrasting a sequence of two commands where the order matters (e.g., "Open the milk carton" and "Pour some milk") with its counterpart – a sequence of two commands where the order doesn't matter (e.g., "Open the package of cereal" and "Pick up the bowl") – may help the students to discern A6–7. The power of contrasting also means that the students' faulty solutions are valuable resources since the teacher may use them to create contrast.

The third strategy is linked to the everyday context of the task Robot Breakfast. Prior research stresses the importance of giving primary students programming activities with an everyday context (Brown & Murphy, 1975; Fivush & Mandler, 1985; Grover & Pea, 2013; Kazakoff & Bers, 2014; Sentance & Csizmadia, 2015). However, our results show how the everyday context becomes a struggle for Leo. It is obvious that he reads the commands from the perspective of his everyday experience of making and eating breakfast. Martin, on the other hand, takes the perspective of the computer scientist (Bell & Lodi, 2019) in terms of considering which commands are available and the meaning of the same. Leo's experiences of eating milk and cereal, and his strong focus on "how it should be [prepared]", seems to prevent him from discerning what Martin discerns. This raises the question of whether a non-everyday context, such as programming a robot to build, for example, a LEGO artefact, would help students to take the perspective of the computer scientist. In such a context, they presumably know that every word in the instructions is important. They also know that the output is a ready-made artefact with a predetermined design, i.e., their own preferences are of no importance. In other words, there are no experiences from their own life that they need to ignore.

The fourth strategy concerns aspects 7 and 12 and the possible advantages of analyzing different sequences that have the same output. During the teacher-led review (Activity 4), the consequences of A7 ("the order of two commands doesn't matter if…") are highlighted when Vincent comments that the two "working" solutions on the whiteboard are not identical. However, the constituted content in terms of aspect A12 – "The order of two commands in a solution that works can change and the result be yet another solution that works" – limits itself to this factual finding. For the students to be able to discern why commands can change position, each sequence needs to be analyzed in relation to its output. Questions like: Which commands change position? How come they can change position without affecting the outcome (Rich et al., 2017)? Are there commands that don't change position? How come? can serve as an entry point to such kinds of analysis. Such questions have the potential to channel attention towards a content that includes A7 and thus enable the students to experience the object of learning in a more complex way.

It is interesting to note that the representation of the students' solutions on the board is a representation of a computational problem implicitly framed by A5–7. Hence, the representation may be used as a computational problem for the students to solve, or in other words, as a tool to mediate said critical aspects. Obviously, it invited Melwin to spontaneously practice computational thinking in terms of discerning a pattern in the representation. Using guiding questions to invite the students to a further analysis of the pattern would therefore be a possible next step. Such an approach would enable the students to further practice computational thinking, and to develop general rules in terms of A5–7.

6.5 Aspects 10-11: Solutions that work or not and improvements of UP activities to qualify aspects 10-11

The lack of focus on the output of the five given commands not only had consequences for the students' possibility to discern A7 and A12. We believe that it also affected the students' possibility to discern why a given sequence could be assessed as a "solution that works", or not, during Activity 4. The constituted content relates to a definition of a working and a non-working solution: something crazy/undesired doesn´t happen/happens (A10-11). However, for the students to be able to discern the meaning of a crazy outcome, and thus to experience the object of learning in a more complex way, they also need to discern and attend to the potentially critical aspect "The meaning of programming is to accomplish something desired and specific". From the perspective of variation theory, a "crazy" outcome can only be discerned if you simultaneously contrast it with its counterpart – a desired output.

One way to direct the students' attention towards the output is to follow up the students' assessments of a sequence with the question: How do you know that it works/doesn't work? Such a question directs attention towards the meaning of programming and thus what the robot is supposed to accomplish. Discerning this aspect is critical for qualifying A10-11 so that it is formulated as "The meaning of a solution that works/doesn't work is that it has/hasn't accomplished the desired output". Nonetheless, there is reason to believe that Martin, Linn and Tim did discern the (only possible) output. From Leo's way of reasoning, we know that he had another (not possible) output in mind. As for Tomas and Rosa, it is difficult to know. This lack of focus on the meaning of programming and the actual output stands in contrast to prior studies' recommendations to implement a practice of predictive thinking (Sentance et al., 2019; Zhang & Nouri, 2019).

Another consequence of the lack focus on the actual output is that the solving of Robot Breakfast may not afford an experience of "thinking like a computer scientist" (Wing, 2006, 2011). For the task to become such an experience requires not only understanding that a sequence must be ordered based on temporal logic (A5). One needs also to understand that the work of the robot should result in a specific and desired outcome, since "coding is an attempt to articulate a precise input to facilitate a particular output"

(Wing, 2011, p. 438). There is reason to believe that students like Leo, Tomas and Rosa would have been helped by a discussion that explicitly focused on the actual output. Knowledge about the meaning of programming in terms of accomplishing a desired output is not intuitively developed through the task, something that is illustrated in Leo's way of directing the outcome towards his own ideas about "how it should be". To conclude, Robot Breakfast invites the students to practice a perspective of the computer scientist (Wing, 2006, 2011), but since it relies on aspects which we cannot expect all students to have discerned, this perspective may not be a lived perspective by every student. In other words, there is reason to believe that not all students experienced doing sequencing as doing sequencing as programming. This therefore implies that students need help to take the perspective of the programmer.

7 Concluding remarks

To conclude, we may infer that the students, to be able to understand the idea of sequencing commands, need to discern several rather detailed aspects. Since the task was seemingly too advanced for three out of the six observed students, we believe that most of the identified aspects are potentially critical for many novice students in primary school. Importantly, one cannot take for granted that these are aspects they discern when observing or interacting with programmable artefacts out of school. Rather, the results imply that that they need to be explicitly dealt with in the primary technology classroom. As confirmed in a series of studies, doing programming activities is far from equal to students learning programming (Bell & Lodi, 2019; Lye & Koh, 2014; Meerbaum-Salant et al., 2010). An important conclusion from our study is that even the most fundamental aspects of the object of learning are by no means intuitive in a task such as Robot Breakfast. However, we believe that the critical aspects we have identified could serve as useful support for teachers who are interested in working with this type of UP assignment. We think that these aspects can help teachers to be aware of what students may need to distinguish already during the planning stage. The critical aspects could also be tested and refined in one or more learning studies (Marton & Fung-Lo, 2007).

Research ethics

Author contributions

Berg, A.: conceptualisation, investigation, methodology, analysis, writing – original draft, writing – review and editing
Axell, C.: conceptualisation, investigation, project and administration, methodology, writing – review and editing
Eriksson, I.: methodology, analysis, writing – original draft, writing – review and editing

All authors have read and agreed to the published version of the manuscript.

Informed consent statement

Informed consent was obtained from all research participants.

Conflicts of interest

The authors declare no conflicts of interest.

References

Ahmed, G., Nouri, J., Zhang, L., & Norén, E. (2020). Didactic methods of integrating programming in mathematics in primary chool: Findings from a Swedish national project. *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*, 261–267. https://doi.org/10.1145/3328778.3366839

Aranda, G., & Ferguson, J. P. (2018). Unplugged programming: The future of teaching computational thinking? *Pedagogika*, *68*(3). https://doi.org/10.14712/23362189.2018.859

Battal, A., Afacan Adanır, G., & Gülbahar, Y. (2021). Computer science unplugged: A systematic literature review. *Journal of Educational Technology Systems*, *50*(1), 24–47. https://doi.org/10.1177/00472395211018801

Bell, T., Alexander, J., Freeman, I., & Grimley, M. (2009). Computer science unplugged: School students doing real computing without computers. *The New Zealand Journal of Applied Computing and Information Technology*, *13*.

Bell, T., & Lodi, M. (2019). Constructing computational thinking without using computers. *Constructivist Foundations*, *14*(3), 342–351.

Bell, T., & Vahrenhold, J. (2018). CS unplugged—How is it used, and does It work? In H.-J. Böckenhauer, D. Komm, & W. Unger (Eds.), *Adventures Between Lower Bounds and Higher Altitudes* (Vol. 11011, pp. 497–521). Springer International Publishing. https://doi.org/10.1007/978-3-319-98355-4_29

Berg, A., & Axell, C. (2023). *Introducing programming in an early primary technology classroom: The distinction between human and robot* (pp. 271–290). https://doi.org/10.1163/9789004687912_013

Bergqvist, E. (2021, June 1). *An inquiry of different interpretations of programming in conjunction with mathematics teaching*. The ninth Nordic Conference on Mathematics Education (NORMA 20), Oslo, Norway.

Brackmann, C. P., Román-González, M., Robles, G., Moreno-León, J., Casali, A., & Barone, D. (2017). Development of computational thinking skills through unplugged activities in primary school. *Proceedings of the 12th Workshop on Primary and Secondary Computing Education*, 65–72. https://doi.org/10.1145/3137065.3137069

Brown, A. L., & Murphy, M. D. (1975). Reconstruction of arbitrary versus logical sequences by preschool children. *Journal of Experimental Child Psychology*, *20*(2), 307–326. https://doi.org/10.1016/0022-0965(75)90106-X

Burke, Q., & Kafai, Y. B. (2012). The writers' workshop for youth programmers: Digital storytelling with scratch in middle school classrooms. *Proceedings of the 43rd ACM Technical Symposium on Computer Science Education*, 433–438. https://doi.org/10.1145/2157136.2157264

Campbell, C., & Walsh, C. (2017). Introducing the "new" digital literacy of coding in the early years. *Practical Literacy: The Early and Primary Years*, *22*(3), 10–12. https://doi.org/10.3316/informit.087120531638806

Chen, P., Yang, D., Metwally, A. H. S., Lavonen, J., & Wang, X. (2023). Fostering computational thinking through unplugged activities: A systematic literature review and meta-analysis. *International Journal of STEM Education*, *10*(1), 47. https://doi.org/10.1186/s40594-023-00434-7

Curzon, P., McOwan, P. W., Cutts, Q. I., & Bell, T. (2009). Enthusing & inspiring with reusable kinaesthetic activities. *Proceedings of the 14th Annual ACM SIGCSE Conference on Innovation and Technology in Computer Science Education*, 94–98. https://doi.org/10.1145/1562877.1562911

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

Faber, H., Wierdsma, 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.

Fadjo, C. L. (2012). Developing computational thinking through grounded embodied cognition. In *ProQuest LLC*. ProQuest LLC.

Fivush, R., & Mandler, J. M. (1985). Developmental changes in the understanding of temporal sequence. *Child Development*, *56*(6), 1437. https://doi.org/10.2307/1130463

Gee, J. P., & Green, J. L. (1998). Chapter 4: Discourse analysis, learning, and social practice: A methodological study. *Review of Research in Education*, *23*(1), 119–169. https://doi.org/10.3102/0091732X023001119

Grover, S., & Pea, R. (2013). Computational thinking in K–12: A review of the state of the field. *Educational Researcher*, *42*(1), 38–43. https://doi.org/10.3102/0013189X12463051

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

Humble, N., Mozelius, P., & Sällvin, L. (2019). *On the role of unplugged programming in K-12 education*. https://doi.org/10.34190/EEL.19.049

Kazakoff, E. R., & Bers, M. U. (2014). Put your robot in, put your robot out: Sequencing through programming robots in early childhood. *Journal of Educational Computing Research*, *50*(4), 553–573. https://doi.org/10.2190/EC.50.4.f

Kullberg, A. (2010). *What is taught and what is learned. Professional insights gained and shared by teachers of mathematics*. https://doi.org/10.13140/RG.2.2.17823.76967

Kullberg, A., Ingerman, Å., & Marton, F. (2024). *Planning and analyzing teaching: Using the variation theory of learning* (1st ed.). Routledge. https://doi.org/10.4324/9781003194903

Lee, J., & Junoh, J. (2019). Implementing unplugged coding activities in early childhood classrooms. *Early Childhood Education Journal*, *47*(6), 709–716. https://doi.org/10.1007/s10643-019-00967-z

Lye, S. Y., & Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12? *Computers in Human Behavior*, *41*, 51–61. https://doi.org/10.1016/j.chb.2014.09.012

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

Mannila, L., & Heintz, F. (2023). *Introducing programming and computational thinking in Grades 1–9: Sweden in an international context* (pp. 60–88). https://doi.org/10.1163/9789004687912_004

Marton, F. (2014). *Necessary conditions of learning* (0 ed.). Routledge. https://doi.org/10.4324/9781315816876

Marton, F., & Fung-Lo, M. L. (2007). Learning from “The Learning Study.” *Tidskrif Foer Laerarutbildning Och Forskning [Journal of Research in Teacher Education]*, *2007*(1), 31–44.

Meerbaum-Salant, O., Armoni, M., & Ben-Ari, M. (Moti). (2010). Learning computer science concepts with scratch. *Proceedings of the Sixth International Workshop on Computing Education Research*, 69–76. https://doi.org/10.1145/1839594.1839607

Pea, R. D. (1986). Language-independent conceptual “bugs” in novice programming. *Journal of Educational Computing Research*, *2*(1), 25–36. https://doi.org/10.2190/689T-1R2A-X4W4-29J2

Rich, K. M., Strickland, C., Binkowski, T. A., Moran, C., & Franklin, D. (2017). K-8 learning trajectories derived from research literature: Sequence, repetition, conditionals. *Proceedings of the 2017 ACM Conference on International Computing Education Research*, 182–190. https://doi.org/10.1145/3105726.3106166

Runesson, U. (2017). Variationsteori som redskap för att analysera lärande och designa undervisning. *Undervisningsutvecklande Forskning*, 45–60.

Sentance, S., & Csizmadia, A. (2015). *Teachers’ perspectives on successful strategies for teaching Computing in school*.

Sentance, S., Waite, J., & Kallia, M. (2019). Teaching computer programming with PRIMM: A sociocultural perspective. *Computer Science Education*, *29*(2–3), 136–176. https://doi.org/10.1080/08993408.2019.1608781

Shute, V., Sun, C., & Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational Research Review*, *22*. https://doi.org/10.1016/j.edurev.2017.09.003

Sivilotti, P. A. G., & Laugel, S. A. (2008). Scratching the surface of advanced topics in software engineering: A workshop module for middle school students. *Proceedings of the 39th SIGCSE Technical Symposium on Computer Science Education*, 291–295. https://doi.org/10.1145/1352135.1352235

Skolverket. (2024). *Läroplan för grundskolan, förskoleklassen och fritidshemmet 2022* (Upplaga 2). Utbildningsdepartementet.

Spektor-Precel, K., & Mioduser, D. (2015a). 5-7 year old Children’s conceptions of behaving artifacts and the influence of constructing their behavior on the development of theory of mind (ToM) and theory of artificial mind (ToAM). *Interdisciplinary Journal of E-Skills and Lifelong Learning*, *11*, 329–345. https://doi.org/10.28945/2332

Spektor-Precel, K., & Mioduser, D. (2015b). *The influence of constructing robot's behavior on the development of theory of mind (ToM) and theory of artificial mind (ToAM) in young children* (p. 314). https://doi.org/10.1145/2771839.2771904

Sun, L., Liu, J., & Liu, Y. (2024). Comparative experiment of the effects of unplugged and plugged-in programming on computational thinking in primary school students: A perspective of multiple influential factors. *Thinking Skills and Creativity*, *52*, 101542. https://doi.org/10.1016/j.tsc.2024.101542

Sung, W., Ahn, J., & Black, J. B. (2017). Introducing computational thinking to young learners: Practicing computational perspectives through embodiment in mathematics education. *Technology, Knowledge and Learning*, *22*(3), 443–463. https://doi.org/10.1007/s10758-017-9328-x

Wainewright, M. (2016). *Lär dig koda: Grunderna i programmering - steg för steg*.

Wing, J. M. (2006). Computational thinking. *Commun. ACM*, *49*(3), 33–35. https://doi.org/10.1145/1118178.1118215

Wing, J. M. (2011, March 6). *Research notebook: Computational thinking - what and why?* Carnegie Mellon School of Computer Science. https://www.cs.cmu.edu/link/research-notebook-computational-thinking-what-and-why

Wohl, B., Porter, B., & Clinch, S. (2015). Teaching computer science to 5-7 year-olds: An initial study with Scratch, Cubelets and unplugged computing. *Proceedings of the Workshop in Primary and Secondary Computing Education*, 55–60. https://doi.org/10.1145/2818314.2818340

Zelazo, P. D., Carter, A., Reznick, J. S., & Frye, D. (1997). Early development of executive function: A problem-solving framework. *Review of General Psychology*, *1*(2), 198–226. https://doi.org/10.1037/1089-2680.1.2.198

Zhang, L., & Nouri, J. (2019). A systematic review of learning computational thinking through Scratch in K-9. *Computers & Education*, *https://authors.elsevier.com/a/1ZIfP1HucdHyVb*, 103607. https://doi.org/10.1016/j.compedu.2019.103607