Informatics in Education, 2025, Vol. 24, No. 1, 77–97

DOI: 10.15388/infedu.2025.03

Entering the Car Park – Primary Pupils' Spontaneous Understanding of Programmed Control Systems in their Daily Life Environment

Gerard DUMMER[1,*], Elwin SAVELSBERGH[1,2], Paul DRIJVERS[2]
[1]*HU University of Applied Science Utrecht, the Netherlands*
[2]*Freudenthal Institute, Utrecht University, Utrecht, the Netherlands*
e-mail: gerard.dummer@hu.nl, elwin.savelsbergh@hu.nl, p.drijvers@uu.nl

Received: July 2024

Abstract. Programmed control systems are ubiquitous in the present-day world. In current educational practice, however, these systems are hardly being addressed, and little is known about children's spontaneous understandings about such systems. Therefore, we explored pupils' understandings prior to instruction in three concrete settings: a car park, an elevator, and an autonomous robot. We analysed written responses from 49 Grade 3 (aged 7 to 10) and Grade 6 pupils (aged 10 to 13) to assess their understandings from two perspectives: the user and the system programmer perspective. Results indicate that most pupils were capable describing programmed systems from a user perspective point of view but found it hard to describe the system programmer perspective. Substantial differences were found between the contexts. The car park context evoked richer descriptions for the user perspective and the system programmer perspective in comparison to the elevator and autonomous robot contexts.

Keywords: physical computing, primary education, programmed control systems programming, user perspective, system programmer perspective.

1. Introduction

Children grow up in an increasingly digitalized world, with programmed systems and devices influencing their lives in visible and invisible ways. Devices such as laptops and smartphones are at the most visible end, but many other applications tend to evade awareness, such as climate control systems, traffic lights, and elevators containing intelligent control (van Keulen, 2010). In fact, such *programmed control systems* are designed to operate as "smooth[ly] as possible and weave themselves into the fabric of everyday life" (Hong, 2017; Satyanarayanan, 2017; Weiser, 1993). In interaction with these systems,

* Corresponding author.

we form socio-technical systems (Boy, 2013, August 26–28) but are usually not aware of their intelligence. Realizing how past technologies changed our society and how these new technologies change our society today, is a prerequisite to make informed decisions about the society we want to live in. Therefore, awareness about such systems, how they operate, and how they weave into our lives is needed for aspiring software designers as well as for consumers and critical citizens (KNAW, 2012; OECD, 2017; SLO, 2015). Moreover, programmed control systems, by virtue of their concrete tangible in- and outputs and their well-defined purposes, can provide fruitful problem-solving contexts for elementary programming education (Garneli *et al.*, 2015, 18–20 March).

Robotics is a specific category of programmed control systems. Education in the field of robotics began in the early seventies with the TORTIS in 1974 (Perlman, 1974) and continues to this day with the latest Lego-robotic kits, such as Lego Mindstorms. Research into the field of robotics in education started in 1975 (López-Belmonte *et al.*, 2021) and continues till this day. Robotics is used as a motivating context to teach basic programming concepts, solve structured and ill-structured problems, and is integrated in other subjects such as science and music (Atman Uslu *et al.*, 2022). The most common use of robotics can be found in the STEM subjects (Anwar *et al.*, 2019; Ouyang, Xu, 2024).

The use of tangible input and output in programming is also known as physical computing. The scope of physical computing soon became larger with the introduction of microcontrollers that had to give an impulse to broaden participation in computing (Blikstein, 2015) in the twenty-first century. This was done by giving more attention to the creative opportunities of microcontrollers (Przybylla & Romeike, 2014) such as e-textiles (Litts *et al.*, 2017, 08 March). Learning about robotics and e-textiles however does not inform children about the programmed control systems they interact with on a daily basis.

Internationally, teachers and researchers are searching for meaningful ways how to address the ever increasingly digitalized world in the primary school curriculum, and there is a need for concrete learning trajectories that work in primary schools (Barr & Stephenson, 2011; Berry, Mike, 2013; Grover & Pea, 2013). In the Netherlands, understanding the technological environment is part of the national curriculum (SLO, 2006), but no connection is made to the programmed world. The other way around, there have been several initiatives to introduce programming in primary education (Jeuring *et al.*, 2016), but in these initiatives no connection is being made to physical computing. Some educational activities, such as the First Lego League, do address physical computing, but they do not connect to technological control systems as they occur in the daily life environment of children.

Various learning theories emphasize the central role of pupils' prior understandings and experiences as a basis for further reasoning and learning. Moreover, from earlier research we know that novice programmers do hold various misconceptions about programming that may interfere with learning if they remain unaddressed (Berry, Michael & Kölling, 2016, 31 March – 03 April; Bower & Falkner, 2015, 27 – 30 January 2015; Du Boulay, 1986; Pea, 1986; Sorva *et al.*, 2013; Žanko, Mladenović, & Krpan, 2023). Therefore, before we can start designing learning trajectories and instructional approaches, we need a clear insight in pupil's prior ideas about programmed control systems.

2. Understanding Programmed Control Systems from Two Perspectives

Understanding a programmed control system can mean different things depending on the perspective one takes. We will distinguish two interrelated perspectives: the user and the system programmer. In order to gain a valid portrayal of a pupil's understanding, both perspectives need to be addressed.

The *user perspective* is grounded in pupils' experiences in daily life and raises expectations about how the system will respond to the user's actions and how interactions with the system will proceed. The user perspective comprises some understanding about the parts of the system (as seen by the user) and about the boundaries of the system (which objects and events are relevant to one's interaction with the system and which are not). The user perspective is important for pupils as a frame of reference to raise expectations about the behaviors of the system and to assess the correctness of their own solutions once they start programming. In previous research about pupils' understanding of technical systems more generally, it was found that pupils do understand that systems consist of parts (input, output, and processing) with different functions, but that they find it hard to set boundaries to the system under consideration, and they cannot explain how the components interact to produce the observed behaviors (Koski & de Vries, 2013).

This brings in the second perspective: the *system programmer perspective*. To become a system programmer of programmed control systems, one needs to understand the inner workings of these systems and know how to program these systems. Systems contain mechanical components such as motors, valves, heating elements, and sensors and a controlling unit that interprets signals from these mechanical parts and, based on their instructions, control these mechanical parts (Mioduser *et al.*, 1996). How these two parts of a system work together is difficult for pupils to understand (Cederqvist, 2022). In this study, we will focus on the controlling unit of specific systems and how pupils think these control systems can be programmed.

In a study about perceptions of the control process, Mioduser, Venezky, and Gong (1996) found that pupils have limited mental models of the computer. They distinguished four levels of understanding: the black box, reactive, switch, and control level. At the black box level, the overall behavior of a system would be described in terms of input and output only. At the reactive level, pupils do have an idea about the use of sensors in the programmed system. At the third level, switch, separate commands-delivering functions appear, making clear that there is a need for a controlling module without further specification of its inner workings. The fourth level, control, is a complete causal model with control specifications included. Mioduser, Venezky, and Gong (1996) provide the following example of a description of a causal model:

> *"We are planning to build a trapdoor that moves from side to side. The door will run on tracks . . . When someone (or something) steps on a part near the door, the door will move over, or if someone breaks the light sensor, the door will open . . . The door can move forward or backward, depending on where the sensor is broken. The door can be*

> *opened from below when someone turns the wheel from the inside... [About the rules governing the doors operation:] IF someone breaks the sensor, then the motor will turn the gear, and the door will open. IF someone turns the wheel, then turn the gear to open the door. IF something does not break the full sensor, then do not open. IF something goes through both sensors within 5 s, then do not open . . .”*

In their study among sixth-grade students, Mioduser *et al.* (1996) found that 26.3% of the descriptions could be classified as black box, 52.6% as reactive, 10.5% as switch, and 0% as control, with the remaining 10.5% giving no relevant descriptions. This confirms that pupils have limited initial knowledge about the fact that the computer controls different processes inside the system.

Pupils have limited initial knowledge about the fact how the computer inside a system controls the system. Are pupils capable making a plan to program the computer for a programmed control system? In a study by Sheehan (2003, July 1–3) children were asked to draw a picture showing a person programming a computer. Younger children (6 or 7 years old) would not know what was meant. Older children (9 and 10 years old) would draw the most ‘technical’ screen layout they could think of, such as a Windows start-up screen, and would equate programming to installing a program. Several other studies confirm that pupils in the lower grades are less likely to talk about programming and programmers compared to pupils in the final grades of primary education (Rücker & Pinkwart, 2016). In a more recent study (Geldreich *et al.*, 2019, October 23–25), it was found that primary school children were aware that devices such as consumer electronics and traffic light needed to be programmed, but they also used the same term to refer to the process of entering map data in a navigation system or to the computer-aided design of a new car. In terms of actions by the programmer, children mentioned a variety of behaviors, ranging from logging in, to combining blocks or creating and transmitting data. From this, it becomes clear that pupils do not have a clear idea about what a programmer does. This is mainly due to the invisible nature of the work of the programmer, who makes sure that the flow of information inside the programmed system is properly working (Hallström & Klasander, 2017). In these studies, pupils were asked what they thought a programmer would do. They were not asked to describe a plan to program the computer.

3. Research Question

Children’s ideas about natural science concepts have been extensively researched, and some research has been done on children’s understandings of robotic systems. These studies show that young children stick to holistic and anthropomorphic interpretations of robot behavior, whereas pupils in upper primary education distinguish more clearly between algorithmic and intentional behaviors (Levy & Mioduser, 2008; Slangen *et al.*, 2011; Van Duuren *et al.*, 1998).

However, although there are relevant similarities, robots are quite different from the control systems children interact with on a daily basis, and much less is known about how children explain the digital world (Geldreich *et al.*, 2019, October 23–25).

Therefore, the aim of this study is to gain insight into what children of different ages, prior to instruction, know about several more and less familiar programmed control systems in their daily life environments. Typically, programming will be taught in the middle and upper grades of primary school. Over these years there is an extensive growth in pupils' cognitive abilities and their knowledge about the world they live in, which may lead to rather different starting points for programming education in the middle grades compared to the upper grades. Therefore, our central research question is:

What prior knowledge do Grade 3 and Grade 6 pupils have about programmed systems in their daily life environment and public space?

In particular, we are interested in the following sub-questions:

1. How do pupils in Grades 3 and 6 describe programmed control systems from a user perspective?
2. How do pupils in Grades 3 and 6 describe programmed control systems from a system programmer perspective?

4. Methods

To answer our questions, we conducted task-based written interviews to assess pupils' understanding about three more or less familiar programmed control systems: the automatic parking lot, the elevator, and the line-following robot. For the selection of programmed systems in real-world situations, we used Van Graft's *et al.*'s (2009) distinctions of real-world situations for science education in primary schools: excursion/holiday, family, healthcare, model building, science inquiry, school, sport, entertainment, traffic, and transport and shopping. These are examples that children encounter in different real-world situations. We also wanted to include a robotic context because robotics is seen as a big idea that captures the essence of science and technology (Dijkgraaf *et al.*, 2008).

Prior to the task-based interviews, the participants and their parents were informed on the goals of the study, and parents signed an informed consent form through the teacher. No formal ethics approval was sought since the data for this study were collected in 2019 before a formal ethics review procedure had been installed in our institution. Stored personal data consists of video recordings, first name, age and sex. This data is stored on a GDPR-compliant medium that provides a sufficient level of data protection.

4.1. *Participants*

Participants were 25 pupils from Grade 3 (aged 7 to 10) and 24 from Grade 6 (aged 10 to 13) in five different primary schools. The schools were located in various urban and semi-urban locations in the Netherlands. Pupils had diverse socioeconomic status. All pupils were native Dutch language speakers. Twenty-five pupils were male (12 in Grade 3 and 13 in Grade 6), and 24 were female (13 in Grade 3 and 11 in Grade 6). Pupils were selected by the teacher to get a representative sample of ability levels. In some classes, pupils already had some programming experience but not in the context

Table 1
Distribution of participants across grades and contexts

Context			Car Park		Elevator		Robot	
Grade			3	6	3	6	3	6
Total # participants			14	12	6	6	5	6
Gender	M	25	7	6	3	3	2	4
	F	24	7	6	3	3	3	2
Experience	S	25	4	7	4	3	3	4
	N	24	10	5	2	3	2	2

Note. Explanation of abbreviations used:
M = male; F = female; S = some; N = none.

of programmed control systems (see Table 1). Ten pupils in Grade 3 had some programming experience and 14 pupils in Grade 6. Pupils in Grade 3 had programming experience with Scratch (mentioned eight times), and one pupil had experience with three programs: Scratch Jr., Lego WeDo, and HTML. In Grade 6 pupils had experience with Scratch (mentioned nine times), Microbit (mentioned three times), Lego Mindstorms (mentioned twice), and Python (mentioned once). Also, two pupils had experience with a 3D printer. Most of the pupils who mentioned Scratch as programming experience had taken only a few introductory lessons to this program.

4.2. *Tasks Used in Interviews*

Parallel tasks were developed for three different contexts: the car park, the elevator, and the line-following robot (see Appendices A, B, and C). The car park and the elevator were supposed to be familiar from daily life. Pupils were not supposed to have much experience with line-following robots in their daily life environment, but these kinds of robots are widely used in educational settings, such as the worldwide educational program FIRST LEGO League. Furthermore, robots were expected to raise high levels of interest among pupils. These contexts, as stated before, also meet the demand for concrete, tangible input and outputs and well-defined purposes. For each of these contexts, one specific scenario was described. For the car park, it was about entering the car park; for the elevator, it was about going up and down a floor; and for the line-following robot, it was to deliver goods in a factory.

4.3. *Procedure*

Each participant was interviewed about one context (see Table 1). Each interview lasted around 20 minutes and took place in a separate room in the school during regular school

hours. First, the participant was asked to describe which actions were needed to enter a car park or to use an elevator (to gain insight into how pupils would describe the user perspective). For the robotic context, the introduction was slightly different because pupils were not expected to have direct experience with robots: first, the pupils were asked where they thought they could find robots and how a robot knew what it had to do, and next they would view two short movies introducing line-following robots in a factory. To gain insight into how pupils would describe the maker perspective, participants were asked to describe the role of the computer in the system under consideration and to describe a plan to program the computer. Questions were read out aloud by the researcher, and pupils wrote their answers in a workbook. To fine-tune the questions and setup, the intervention was pilot tested with four pupils and adjusted accordingly. For the specific questions, see Appendices A, B, and C.

4. *Data Analysis*

The first part of the interview was aimed at the user perspective; the second part was about the system programmer perspective. Pupils' written documents were analyzed per perspective and per context using the qualitative data analysis program Atlas.ti. To analyze the description of the user perspective, we used the labels input, output, processing, and boundary as defined by Koski and De Vries (2013). For instance, if a pupil would mention arrive at the gate, push the button, take a ticket, gate opens, drive through the gate, gate closes, find a parking lot, go shopping. These objects would be labelled respectively: action outside the boundary of the system (for short, out of boundary), input, output, output, input, output, out of boundary, and out of boundary.

For the system programmer perspective we coded and analyzed the descriptions for the way pupils described the system and how they described the processing steps in any form including the use of input and output. Answers were labeled as blank (pupils did not write anything), the not-system programmer-perspective and system programmer perspective. The descriptions were labeled as not-system programmer perspective if pupils described inputs and outputs from the user point of view, if they would put the computer in the first-person perspective without adding any new information to the user perspective scenario, or if the pupils did not describe a plan to program the computer but the action of the programmer while programming. Descriptions for the system programmer perspective were labeled as black box (only descriptions of input and output), reactive (input, output, and descriptions of sensors), switch (input, output, sensors, and descriptions of a controlling unit) or control (description of a complete causal model).

To assess the inter-rater reliability of our coding, all data were independently coded by two coders, and Cohen's Kappa was calculated as a measure of agreement. For the user perspective this resulted in a Cohen's kappa of 0.80 (based on 39 data units); for the system perspective, we calculated a Cohen's kappa for questions about the computer and for questions about a plan to program the computer. For questions about the computer, this resulted in a Cohen's kappa of 0.66 (based on 48 data units), and, for questions about

a plan to program the computer, this resulted in a Cohen's kappa of 0.79 (based on 46 data units). After this, the two coders discussed their differences until agreement was reached. This consensus coding was used in all further analyses.

5. Results

We present our results along the lines of the two perspectives: user and system programmer. For each perspective, we indicate the differences between Grade 3 and Grade 6. As we did not find differences between pupils with and without programming experiences, we do not present our findings between those two kinds of groups of pupils.

5.1. *The User Perspective*

Pupils were asked to describe the user perspective for the car park and the elevator. We did not ask pupils on the user perspective or the line-following robot context because we expected them not to have any experiences as a user with such a robot. We did ask pupils where they thought they could encounter a robot. Table 2 shows that the answers from the user perspective for the car park consisted of descriptions of the input, output, processing, and boundary. Out of the 26 pupils interviewed about "entering a car park", one pupil had no experience with car parks at all and could not answer any of the questions.

For the car park, pupils could describe input, output, and processing in different ways. Input was described as: push a button, put ticket in the machine, buy ticket, enter debit card, open barrier with ticket, scan ticket. Output was described as follows: receive ticket, barrier opens, and barrier closes. Processing was mentioned in terms of the presence of a computer and calculating costs. Also, different kinds of actions outside the boundary of the control system were given: drive to the barrier, drive through the car park, exit the car park, follow arrows on the ground in the car park, stop car, open window.

As we zoom in onto pupils' descriptions of the user perspective for the car park, it is interesting to notice that, although the question was about entering the car park, most answers also contain elements usually associated with leaving the car park. Some pupils first describe entering and then leaving the car park, like these Grade 3 and 6 pupils:

Table 2
Classification of answers on the user perspective for the car park

	Grade 3	Grade 6	Total
Input	11	12	23
Output	16	16	32
Processing	1	1	2
Boundaries	8	8	16

> *I drive up the hill, and then up there is a barrier. Next to the barrier is a computer. You must push a button. If you push the button, you get an entrance ticket. With this ticket, you can leave the car park* (Grade 3 pupil).

> *You must push a button. Then you get a ticket, and then the barriers open, and if you leave you must go to the pay terminal, and then you have to put money into it and you have to buy a ticket. Then you have to hand in your ticket, and the barrier opens* (Grade 6 pupil).

Others seem to mix up elements from both processes:

> *You must pay first, then you receive a ticket. You put that ticket in your car* (Grade 3 pupil).

Not all answers were as extensive as the ones presented above. Other pupils described the entrance of the car park as a simple event. A typical example of such an answer would be:

> *You put a ticket in the thing and then the barrier opens* (Grade 3 pupil).

The diversity of descriptions for the car park contrasts with the results for the elevator context. In this context, descriptions were mostly limited to input: pushing a button to call the elevator and pushing a button to go to a specific floor. Only one Grade 3 pupil described an output (the doors of the elevator open), and four Grade 6 pupils described output: opening the doors and movement of the elevator from another floor to the caller. None of the pupils mentioned aspects of processing (presence of a computer or system) or actions beyond the boundaries of the system (for example, walking towards the elevator).

None of the Grade 3 pupils had a clear idea where robots worked or where they could meet a robot. One pupil said he knew robots from the movies and that if something went wrong, a robot could turn evil. All pupils from Grade 6 knew where they could meet robots: in places difficult to reach for humans (other planet, under water, or near a volcano), for cleaning, answering questions at a reception, moving heavy loads, in a nursing home to care for the elderly, to play soccer, and as an assistant for scientists. They described the functions of the robot without mentioning any specific user interactions.

From these results, we can conclude that pupils are fairly capable describing the user perspective of the car park context but give a limited description of the elevator and robot context.

5.2. *The System Programmer Perspective*

Pupils were asked to describe the system programmer perspective for the car park, elevator, and line-following robot. They had to answer questions about the role of the

computer in the system and if they could make a plan to program the computer. First, we will describe for each context the answers pupils gave for the role of the computer in the system. Second, we will describe for each context the answers pupils gave about the plan to program the computer. Table 3 gives an overview of all the answers for the three contexts for questions about the role of the computer.

To describe the role of the computer in the car park context we asked pupils two questions. For the first question, pupils had to imagine that they were the computer of the car park, and they had to take care of cars entering and exiting the car park. In the first question, a car would arrive at the gate, and the button would be pressed to request admission. In the second question, the situation had the added condition that there would be only eight spots in the car park, and the computer would have to ensure that no more than eight cars get admitted (see Appendix A).

From Table 3, it becomes clear that almost half of the Grade 3 pupils gave answers that were categorized as not-system programmer perspective. Most Grade 3 pupils who answered this question from a computer perspective described it as a black box. Almost all Grade 6 pupils described question one from a system programmer perspective point of view. These descriptions were categorized as black box descriptions, reactive, and switch. None of the pupils gave a control description. A typical not-system programmer perspective answer from a Grade 3 pupil is:

> *If you push the button, a ticket comes out. So, if you pushed the button a ticket comes out.*

A typical black box description from a Grade 3 pupil is:

> *If I were a computer, they had to pay first, and then I would open the barrier.*

In both examples, there seems to be a person in the description. In the first description, the 'you' refers to a real person. The 'I' in the second description refers to the per-

Table 3

Classification of answers to the role of the computer questions for the car park, elevator, and robot context

Grade	Car Park				Elevator				Robot			
	Q1		Q2		Q1		Q2		Q1		Q2	
	3 (n=14)	6 (n=12)	3 (n=14)	6 (n=12)	3 (n=6)	6 (n=6)	3 (n=6)	6 (n=6)	3 (n=5)	6 (n=6)	3 (n=5)	6 (n=6)
Blank	2	0	2	0	0	0	0	0	2	0	0	0
Not-system programmer perspective	5	1	1	1	0	0	0	0	0	0	0	0
System programmer-perspective	7	11	11	11	6	6	6	6	3	6	5	6
Black box	4	4	11	4	6	6	6	6	3	6	5	6
Reactive	2	4	0	5	0	0	0	0	0	0	0	0
Switch	1	3	0	2	0	0	0	0	0	0	0	0
Control	0	0	0	0	0	0	0	0	0	0	0	0

son as a computer. Therefore, the first example is categorized as not-system programmer perspective and the second example is categorized as system programmer.

Three answers from Grade 6 pupils were labeled switch, such as the following example:

> *If the button is pressed, then the computer knows what time it is. This is remembered well, and later, when you are back and the ticket is pushed into the entrance of the machine, the computer knows how long you have been in the car park. Then it calculates how expensive it is. They pay, and the computer opens the barrier.*

This answer is described as switch because there is explicit mention of control processes inside the computer. It does not qualify as a causal description because the control specifications are not included.

For the second role of the computer question about the car park, almost all pupils answered this from a system programmer perspective point of view. Grade 3 pupils' descriptions were all labeled as black box descriptions. Descriptions from Grade 6 pupils varied but could mainly be categorized as reactive. A typical example of a Grade 3 pupil that was labeled as black box was the following:

> *I would not take the card and say that they must go to a different car park.*

A typical answer from a Grade 6 pupil that was labeled reactive:

> *1. Count the cars that park*
> *2. If there are 8, then do not open the barrier*
> *3. If cars leave, then more cars can get in.*

This answer is labeled reactive because counting the cars implies the presence of a kind of sensor.

To describe the role of the computer in the elevator context, pupils were also presented with two questions. The first question was about a scenario where the elevator was being called to the ground floor, so there was no issue about the direction the elevator should go. The second question was about a scenario where the elevator was being called to the middle floor, which is potentially more complex since the direction to go depends on the current location of the elevator (see Appendix B). All answers from Grade 3 and Grade 6 pupils could be categorized as black box answers. For the first question, all pupils from Grade 3 saw the computer as a physical object that went up and down with the elevator. An example of this is the next quote:

> *If he or she pushes the button, then I will go to the floor* (Grade 3 pupil).

All Grade 6 pupils described the role of the computer, and they did so mostly by describing the output.

To describe the role of the computer in the robot context, pupils were shown two short movies. In these movies, pupils saw line-following robots delivering packages in a factory. Pupils were asked which steps the computer of the robot had to take to make sure that the robot collects the packages and delivers them. In this context, the answers

were also not as diverse as in the car park context, and all answers could be categorized as black box. Two pupils in Grade 3 did not know how to answer this question. Notable differences between Grade 3 and Grade 6 pupils were that Grade 3 pupils answered this question stating what the computer had to do (for instance, the computer controls the robot) and Grade 6 pupils mentioned how the computer would do this (for instance, following the black line; signaling if there is something on the robot).

Following the questions about the role of the computer in each system, pupils were asked to note precisely the plan to program the computer. We categorized the answers in blank, not-system programmer perspective, and system programmer perspective (black box, reactive, switch, and control). In Table 4, the frequency distribution of the answers is summarized.

As can be seen in Table 4 seven pupils in Grade 3 made a description that was labeled not-system programmer perspective. They described what the programmer was doing instead of what the plan to program a computer looked like. An example is presented in Fig. 1, showing the programmer sitting behind his desk to program the computer. None of the pupils in Grade 6 gave this kind of answers.

We invited pupils to write or draw the plan to program the computer. This resulted in descriptions and drawings of the plan. Most of these descriptions and drawings were labeled as black box. An example of a black box description of a plan is the following from a Grade 3 pupil in the elevator context:

> *If a person wants to go to another floor, I first check which floor, and then I check if there are no other persons who want to go also with the elevator.*

Across all contexts and in all grades, pupils gave these kinds of descriptions.

Pupils drew different kinds of drawings to make the plan to program the computer clear. All drawings were labeled as black box. One kind of drawing depicted only a moment in the situation of the context. No clear sequence or logic could be distinguished. An example of a Grade 3 pupil in the car park context can be seen in Fig. 2.

Another kind of drawing that occurred more sparsely (only three pupils wrote an answer in this for) were schematic drawings. An example by a Grade 3 pupil is presented in Fig. 3.

Table 4

Classification of Answers about the Plan of the Programmer Questions for each of the Three Contexts

	Car park		Elevator		Robot	
Grade	3 (n=14)	6 (n=12)	3 (n=6)	6 (n=6)	3 (n=5)	6 (n=6)
Blank	2	2	0	1	3	1
Not-system programmer perspective	7	0	0	0	0	0
System programmer perspective	5	10	6	5	2	5
Black box	5	5	6	3	2	5
Reactive	0	0	0	0	0	0
Switch	0	5	0	2	0	0
Control	0	0	0	0	0	0

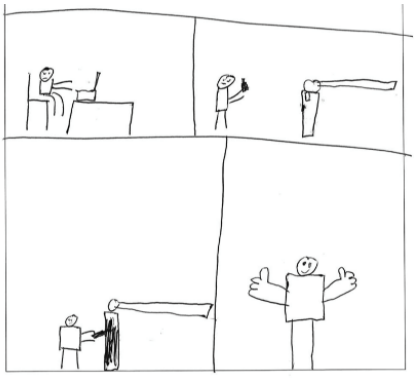

Fig. 1. Programmer at work to program the car park (Grade 3).

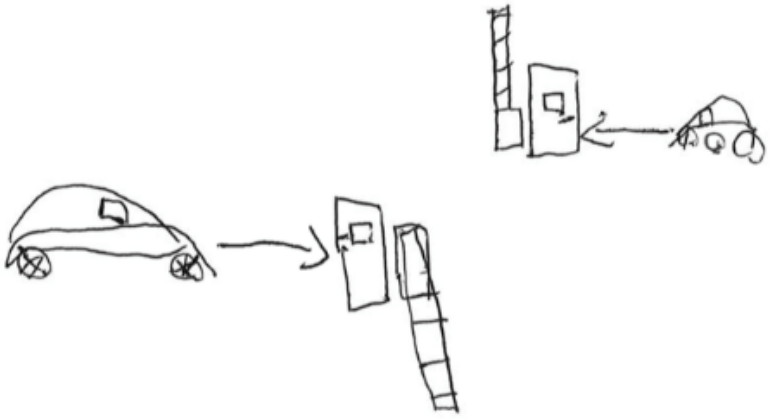

Fig. 2. Situational drawing without any explanation from a Grade 3 pupil.

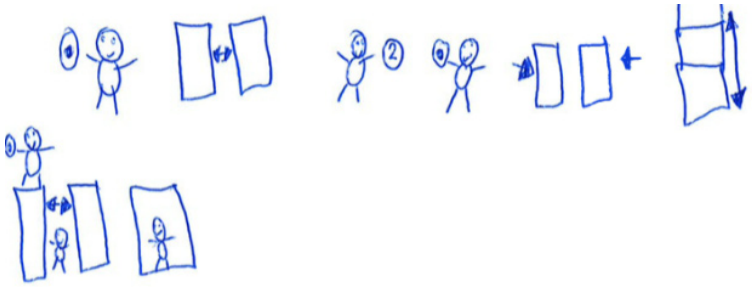

Fig. 3. Example of a schematic drawing in the elevator context (Grade 3 pupil).

No descriptions or drawings were labeled as reactive or control. Seven descriptions were labeled as switch. Fig. 4 presents an example of a switch description in the car park context.

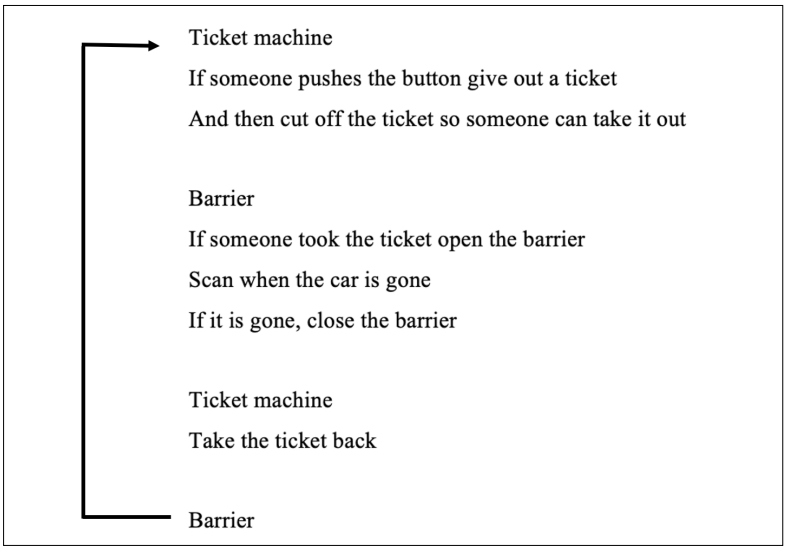

Fig. 4. Detailed Sequential Description of Grade 6 Pupil of the Elevator Context.
(The arrow back to the top is meant to indicate the process starts again for the next car.)

The following is an example of a switch description from a Grade 6 pupil in the elevator context:

> *When the button is pressed*
> *You must go to the floor where [the button] is pressed.*
> *Then you open the door and wait until a floor is selected and you close the doors.*
> *Go to the selected floor.*
> *If on the way [to the selected floor] a button is pushed, you stop and wait until a floor is selected and you close the door and go on to the floor of the previous [person].*
> *Open the doors, wait, and close them again.*
> *Go to the floor of the next [person] and open the doors, wait, and close them again.*

6. Conclusion and Discussion

The central research question for this study was: What prior knowledge do Grade 3 and Grade 6 pupils have about programmed systems in their daily life environment and public space? Our first sub-question was about the ways pupils describe programmed control systems in terms of input, output, processing, and boundaries from the user perspective. We found a clear difference between the car park and the elevator in that the car park yielded more elaborate descriptions. This notable difference may be due to the fact that the interactions with the car park are more salient, and the actual output is

more visible compared to the elevator. Systems that provide visible output may be more accessible for learners to reason about. Because robots did not occur in the daily lives of the pupils, we could not ask similar questions about robots (which have also more visible output), and therefore, we cannot say anything about the differences between these contexts. The descriptions of the car park included descriptions that could be categorized as out of boundary actions. This is in line with finding from earlier research that pupils find it hard to set boundaries to a system (Koski & de Vries, 2013). We did not find clear differences between Grade 3 and Grade 6 pupils describing the system from a user point of view. We conclude that almost all pupils have sufficient experience with these kinds of programmed control systems to build upon for studying the inner workings of the system.

With regard to the second sub-question about the system programmer perspective, we asked questions about the role of the computer and the plan to program the computer. We saw clear differences between Grade 3 and Grade 6 pupils regarding the role of the computer questions. Grade 3 pupils sometimes gave answers that were similar to answers given in the user perspective, and when they did describe what the role of the computer was, it remained a black box to them. Answers of Grade 6 pupils could be categorized more frequently as reactive or switch. None of the pupils gave a description that could be categorized as control. There is also a notable difference between the contexts. The car park evoked more elaborate descriptions of the role of the computer compared to the elevator and line-following robot contexts.

Questions about the plan to program the computer resulted in similar results. Most Grade 3 pupils gave descriptions that could be categorized as black box. Grade 6 pupils managed to make descriptions that could be categorized as switch. Only in the robot context, Grade 6 pupils did not manage to make descriptions beyond the level of black box.

When we compare the results of the role of the computer questions to those about the plan to program the computer, it becomes clear that for Grade 3 pupils, the additional question yielded only a few additional insights. However, Grade 6 pupils, who also answered questions about the role of the computer mostly as black box or reactive, were able to demonstrate additional insight in response to the question about the plan to program the computer.

As a limitation of this study, it should be noted that pupils were asked to write or draw their answers, with no room for follow-up questions. Consequently, if pupils did not show a particular insight in our assessment, we cannot be sure that they understood the question or were unable to show their insight. Rather, our findings should be taken as a lower threshold on what pupils may think of more or less spontaneously when reasoning about these particular control systems.

We wrote that robotics is seen as a motivating context to teach basic programming concepts, solve structured and ill-structured problems, and is integrated in other subjects such as science and music (Atman Uslu *et al.*, 2022). Based on our findings in this study, we would argue that robotics does not always provide the most insightful context to learn programming concepts. If pupils do not understand how robots work from the inside, other physical computing contexts that are more familiar may be more suitable to learn basic programming concepts.

What are the implications for developing a lesson series? In this study, pupils had to describe the system from a user perspective and a system programmer perspective. Pupils were fairly capable describing the system from a user perspective. The better pupils can describe the way they use a system, the easier it seems to change their perspective to the system programmer perspective. The most elaborated user descriptions were found in the car park context. This context also provoked the most comprehensive descriptions of the system programmer perspective. The car park context seems to meet two conditions that contribute to these rich descriptions: pupils were familiar with the system, and the car park has concrete, visible, tangible inputs and outputs. The robot context does not meet the first condition: most pupils were not familiar with robots in their immediate environment. The elevator context does not meet the second condition: the input and output are not visible for pupils.

What does this mean for developing lesson series? Describing the user perspective is a good starting point to discover the inner workings of a digital control system. Digital control systems that are familiar and have concrete, visible, tangible inputs and outputs are easier to use than systems that do not meet these two conditions.

Although systems that are familiar and have concrete, visible, input and output are easier to start with, that does not mean pupils can make systematic descriptions of the way they use these systems. The user perspective descriptions of the car park were elaborated but not systematic. Pupils mixed up processes that belonged to entering the car park and leaving the car park. Pupils still need help to make systematic descriptions of the user perspective so that it can help them to better understand the system programmer perspective.

Digital control systems that do not meet these two conditions can also be used, but pupils need more help to fully understand the inner workings of the digital control system at hand. In this study, the robot did not meet the first condition, but educational robot kits are widely used to teach programming. The elevator does not fully meet the second condition but is familiar and simple enough to explain to pupils in primary school.

Acknowledgments

We would like to thank the members of the Research group Curriculum Development in Primary and Secondary Education of the HU University of Applied Science Utrecht for their critical contributions developing the task-based interviews. A special thanks to Geertje Damstra for coding the data with us. Furthermore, we would like to thank all teachers and pupils who participated in this study.

References

Anwar, S., Bascou, N. A., Menekse, M., & Kardgar, A. (2019). A systematic review of studies on educational robotics. *Journal of Pre-College Engineering Education Research (J-PEER), 9*(2), 2.

Atman Uslu, N., Yavuz, G. Ö, & Koçak Usluel, Y. (2022). A systematic review study on educational robotics and robots. *Interactive Learning Environments*, 1–25.

Barr, V., & Stephenson, C. (2011). Bringing computational thinking to K-12: what is Involved and what is the role of the computer science education community? *Acm Inroads, 2*(1), 48–54.

Berry, M., & Kölling, M. (2016, 31 March–03 April). Novis: A notional machine implementation for teaching introductory programming. Paper presented at the *2016 International Conference on Learning and Teaching in Computing and Engineering (LaTICE),* 54–59.

Berry, M. (2013). *Computing in the national curriculum: A guide for primary teachers. Computing at School*. Computing at School.

Blikstein, P. (2015). Computationally Enhanced Toolkits for Children: Historical Review and a Framework for Future Design. *Foundations and Trends in Human-Computer Interaction, 9*(1), 1–68.

Bower, M., & Falkner, K. (2015, 27– 30 January 2015). Computational Thinking, the Notional Machine, Preservice Teachers, and Research Opportunities. Paper presented at the *Ace,* 37–46.

Boy, G. A. (2013, August 26–28). From STEM to STEAM: toward a human-centered education, creativity & learning thinking. Paper presented at the *Proceedings of the 31st European Conference on Cognitive Ergonomics,* 1–7.

Cederqvist, A. (2022). Designing and coding with BBC micro: bit to solve a real-world task–a challenging movement between contexts. *Education and Information Technologies*, 1–35.

Dijkgraaf, R., Fresco, L., Gualthérie van Weezel, T., & Van Calmthout, M. (2008). *De bètacanon*. Meulenhoff.

Du Boulay, B. (1986). Some difficulties of learning to program. *Journal of Educational Computing Research, 2*(1), 57–73.

Garneli, V., Giannakos, M. N., & Chorianopoulos, K. (2015, 18–20 March). Computing education in K-12 schools: A review of the literature. Paper presented at the *2015 IEEE Global Engineering Education Conference (EDUCON),* 543–551.

Geldreich, K., Simon, A., & Starke, E. (2019, October 23–25). Which Perceptions Do Primary School Children Have about Programming? Paper presented at the *Proceedings of the 14th Workshop in Primary and Secondary Computing Education,* 1–7.

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

Hallström, J., & Klasander, C. (2017). Visible parts, invisible whole: Swedish technology student teachers' conceptions about technological systems. *International Journal of Technology and Design Education, 27*(3), 387–405.

Hong, J. (2017). The privacy landscape of pervasive computing. *IEEE Pervasive Computing, 16*(3), 40–48.

Jeuring, J., Corbalan, G., van Es, N., Leeuwestein, H., & van Montfort, J. (2016). *Leren programmeren in het PO–een literatuurreview* [Learning to program in primary education – a literature review]. `https://www.nro.nl/sites/nro/files/migrate/003-en-036-Leren-programmeren-in-het-PO-een-literatuurreview.pdf`

KNAW. (2012). *Digitale geletterdheid in het voortgezet onderwijs* [Digital literacy in secondary education]. `https://www.knaw.nl/nl/publicaties/digitale-geletterdheid-het-voortgezet-onderwijs`

Koski, M., & de Vries, M. (2013). An exploratory study on how primary pupils approach systems. *International Journal of Technology and Design Education, 23*(4), 835–848.

Levy, S. T., & Mioduser, D. (2008). Does it "want" or "was it programmed to..."? Kindergarten children's explanations of an autonomous robot's adaptive functioning. *International Journal of Technology and Design Education, 18*(4), 337–359.

Litts, B. K., Kafai, Y. B., Lui, D., Walker, J., & Widman, S. (2017, 08 March). Understanding high school students' reading, remixing, and writing codeable circuits for electronic textiles. Paper presented at the *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education,* 381–386.

López-Belmonte, J., Segura-Robles, A., Moreno-Guerrero, A., & Parra-González, M. (2021). Robotics in education: a scientific mapping of the literature in Web of Science. *Electronics, 10*(3), 291.

Mioduser, D., Venezky, R. L., & Gong, B. (1996). Students' perceptions and designs of simple control systems. *Computers in Human Behavior, 12*(3), 363–388.

OECD. (2017). *OECD Skills Strategy Diagnostic Report: The Netherlands 2017*. OECD Publishing. 10.1787/9789264287655-en

Ouyang, F., & Xu, W. (2024). The effects of educational robotics in STEM education: A multilevel meta-analysis. *International Journal of STEM Education*, *11*(1), 7.)

Pea, R. D. (1986). Language-independent conceptual "bugs" in novice programming. *Journal of Educational Computing Research, 2*(1), 25–36.

Perlman, R. (1974). TORTIS (Toddler's Own Recursive Turtle Interpreter System).

Przybylla, M., & Romeike, R. (2014). Physical Computing and Its Scope – Towards a Constructionist Computer Science Curriculum with Physical Computing. *Informatics in Education, 13*(2), 241–254.
Rücker, M. T., & Pinkwart, N. (2016). Review and discussion of children's conceptions of computers. *Journal of Science Education and Technology, 25*(2), 274–283.
Satyanarayanan, M. (2017). The emergence of edge computing. *Computer, 50*(1), 30–39.
Sheehan, R. (2003, July 1–3). Children's perception of computer programming as an aid to designing programming environments. Paper presented at the *Proceedings of the 2003 Conference on Interaction Design and Children,* 75–83.
Slangen, L., van Keulen, H., & Gravemeijer, K. (2011). What pupils can learn from working with robotic direct manipulation environments. *International Journal of Technology and Design Education, 21*(4), 449–469.
SLO. (2006). *Kerndoelen primair onderwijs* [Core objectives primary education]. SLO.
SLO. (2015). *Voorbeeldmatig leerplankader computational thinking* [Exemplary Computational Thinking Curriculum Framework] http://curriculumvandetoekomst.slo.nl/21e-eeuwse-vaardigheden/digitale-geletterdheid/computational-thinking/voorbeeldmatig-leerplankader
Sorva, J., Karavirta, V., & Malmi, L. (2013). A review of generic program visualization systems for introductory programming education. *ACM Transactions on Computing Education (TOCE), 13*(4), 1–64.
Van Duuren, M., Dossett, B., & Robinson, D. (1998). Gauging children's understanding of artificially intelligent objects: a presentation of "counterfactuals". *International Journal of Behavioral Development, 22*(4), 871–889.
Van Graft, M., Boersma, K. T., Goedhart, M., Van Oers, B., & De Vries, M. (2009). *De concept-contextbenadering in het primair onderwijs* [The concept-context approach in primary education]. SLO
Van Keulen, J. (2010). *Wetenschap & Techniek: IJkpunten voor een domein in ontwikkeling* [Science & Technology: Benchmarks for an evolving domain]. Platform Bèta Techniek.
Weiser, M. (1993). Some computer science issues in ubiquitous computing. *36*(7), 75–84. https://doi.org/10.1145/159544.159617
Žanko, Ž., Mladenović, M., & Krpan, D. (2023). Mediated transfer: impact on programming misconceptions. *Journal of Computers in Education*, *10*(1), 1–26.

G. Dummer is a teacher educator for primary schools and PhD-candidate in the Curriculum Development research group at the HU University of Applied Sciences. His research focuses on the development of learning pathways for physical computing in the upper grades of primary education.

E. Savelsbergh is a professor of curriculum research. Central questions in his research program include: what knowledge and skills do pupils need in order to flourish, and to make a meaningful contribution to society in society; how can these subjects be taught in motivating and coherent ways; and what knowledge and skills do teachers need to achieve this?

P. Drijvers is full professor in mathematics education at the Freudenthal Institute of Utrecht University's Science Faculty. His research interests include the role of ICT in mathematics education, mathematical thinking, and embodiment in mathematics education.

Appendix A
Interview setup car park context / problem / situation

User perspective

I just asked if there are devices in your house containing a computer. If you go outside, you will also encounter systems containing a computer. For example, when you ride your bike or if you are in the car with someone.

Maybe you have been in a city with the car and the car had to be parked in a car park or garage.

- Have you ever been in a car park in which you had to pay?
- Was it possible to drive straight on to the car park?
- Can you describe what you had to do, to enter the car park?

System programmer perspective

Question 1 about the role of the computer

I just asked what you had to do to enter the car park. Now I like to know what you think which steps the computer of the car park must take to take to ensure you can enter the car park. Imagine that you are the computer of the car park, and you must take care of cars entering and exiting the car park. Which steps would you take as a computer? You start as a computer the moment someone pushes the button to lift the barrier of the car park.

Question 2 about the role of the computer

There are only 8 spots on the car park from which you are the computer. Which steps would you take, as a computer, to ensure that no more than 8 cars can be parked in the car park?

Question to make a plan to program the computer

You were just asked to be the computer of the car park. Now we will assign you with another role. Namely that of the programmer. It's your task to note precisely what the computer of the car park has to do. The cars have to enter and exit the car park. You also have to tell the computer that the car park has only eight spots for the cars. I am a stupid computer, so you must note precisely what I have to do. You can write it down. You can also draw arrows, blocks, and lines to make clear as a programmer what you want the computer to do.

Appendix B
Interview setup elevator situation

User perspective
I just asked if there are devices in your house containing a computer. If you go outside, you will also encounter systems containing a computer. For example, if you have to go somewhere else. Maybe you have been to a building in which there was an elevator?

- Did you ever take the elevator?
- What did you have to do when you stood before the elevator?
- What did you have to do when you were in the elevator?

System programmer perspective
Imagine you are the computer of the elevator. Your task is to make the elevator go up and down to the right floor. There is someone one the first floor, pushing a button to make the elevator go to him.

Question 1 about the role of the computer
What would you have to do as a computer to get the elevator to the right floor?

Question 2 about the role of the computer
There is a building with a ground floor, first floor and second floor.

The elevator is on the first floor. There is someone on the second floor who wants to take the elevator.
Which instructions can the computer give to the elevator?

Question to make a plan to program the computer
You were just asked to be the computer of the elevator. Now we will assign you with another role. Namely that of the programmer. It's your task to note precisely what the computer of the elevator has to do. You have to make sure that people can get in and can go to a specific floor. I am a stupid computer, so you have to note precisely what I must do. You can write it down. You can also draw arrows, blocks, and lines to make clear as a programmer what you want the computer to do.

Appendix C
Interview setup robot

User perspective
One of the devices containing a robot is a computer. You can encounter robots in all sorts of places. Robots are devices that can move and are programmed. They can take over a task or assist people with a task.

- Where do you think, you can encounter a robot?
- Does a robot know from itself what it must do?
- How would a robot know this?

System programmer perspective
We are going to look at two movies of robots delivering packages in the factory.
Pupils look at a movie of an Amazon warehouse where robots deliver packages from one side of the factory to the other (`https://youtu.be/cLVCGEmkJs0?t=29`) and of an automated guided vehicle called Weasel in the Bachmann Forming factory (`https://youtu.be/aP6k5VYvGHc?t=8`).

Question 1 about the role of the computer
What would the computer of the robot do if someone pushes the button, or the sensors detects something?
You have just seen two movies of robot delivering goods in the factory. In the last movie you saw a robot collecting a box and deliver it somewhere else.

Question 2 about the role of the computer
Which steps has the computer of the robot to make to make sure that the robot collects the box and delivers it.

Question about the plan to program the computer
You were just asked to be the computer of the robot. Now we will assign you with another role. Namely that of the programmer. Your assignment is to design a computer program for the robot.
The computer program must take care of the following:

- The robot can follow a black line
- The robot stops on a specific spot to collect packages.
- The robot stops on a specific spot to deliver the packages.

How would you make a plan for such a computer program?