Signs of learning in middle school computing education

1st Author Susanne Kjällander
Department of Child and Youth Studies
Stockholm University
Stockholm, Sweden
susanne.kjallander@buv.su.se

2nd Author Anna Åkerfeldt
Department of teaching and learning
Stockholm University
Stockholm, Sweden
anna.akerfeldt@su.se

3rd Author Linda Mannila
Department of Computer and Information Science
Linköping University
Linköping, Sweden
linda.mannila@liu.se

4th Author Fredrik Heintz
Department of Computer and Information Science
Linköping University
Linköping, Sweden
fredrik.heintz@liu.se

Abstract
Programming has been part of Swedish elementary school curriculum for six years and the aim of this full paper is to find out how teachers can design programming activities so that students engage and learn. A mix-methods research project with a social semiotic, multimodal theoretical framework – Designs for learning – is used to investigate teaching and learning in a class during three years. The results in this small-scale study indicate that collaboration is a successful didactic design for programming lessons in school. Computational thinking is prevalent and both digital skills (such as coding) and digital competencies (such as understanding the impact of technology in society) are practiced and met in programming lessons merging Science, Technology, Engineering, Arts, and Mathematics.

***Keywords** – programming, coding, digital, K12 education, designs for learning, multimodal*

I. Introduction

Programming and computational thinking [1] are new in many countries' curriculums. Most studies in a research review of almost 30 projects lean on a view of knowledge and learning where students create something when they program – this in turn consolidates what they have learned [2]. A study [3] show that young pupils showed motivation and enthusiasm for learning when they programmed and a lot of research suggests that younger children find it easier to learn programming by using visual or physical resources [4, 5]. Digital competence and programming have been part of the Swedish elementary school curriculum since 2018 [6]. In especially Math, students are expected to train computational thinking [1] in programming activities. This paper aims to understand what supports students learning in programming. The social semiotic, multimodal theoretical framework [7, 8] Designs for learning [9] is used to investigate potential signs of learning in pupils´ multimodal representations when they for example write code or make use of block programming in the primary and secondary transformation unit [10]. Aligned prior research [2] the article will show how students engage in computational thinking while creating, often with aesthetic expressions. Their learning is consolidated into knowledge when they use their new knowledge in advanced ways in new situations. Earlier research [3] concerning motivation and enthusiasm for learning is confirmed and a majority of the lessons studied make use of visual resources which is both in this study, and in prior research, understood as and easier, or more attractive, way to learn programming [4, 5] and collaboration works well. Teachers and students learn programming together (cf. "twin imperatives of technology" [11]) and they help and collaborate with each other. The article will illustrate how visual modes and multimodal narratives are appreciated as affordances by the students. It will also show students´ signs of learning while they form, transform and transduce their new knowledge in programming by means of aesthetics in advanced ways in new situations. These new situations are often framed by Science, Technology, Engineering, Arts, and Mathematics: STEAM [12].

Earlier research by Karlsohn [13] argues that there was previously a concern that technology would threaten democracy. However, recent research by Samuelsson [14] shows that the concern has now been replaced by an understanding that digital competence is a prerequisite for participation and influence in society. Hernwall [15] shows that the school's digitalization is conceptualized as an opportunity to support specific competences or as a threat to communication/well-established ways of performing education.

Digitalization means changes in life and society. According to the Swedish government [6] these changes demand that school gives all children the opportunity to develop their digital competence, in a way that emphasizes equity. These changes can be summarized as follows: 1) programming is introduced as a prominent feature in different subjects in primary education, especially in Technology and Math, 2) students will be strengthened in their critical evaluation of sources, 3) students can solve problems and translate ideas into practice in a creative way, 4) students should work with digital texts, media and tools, 5) students should use and understand digital systems and services and 6) students should develop an understanding of the impact of digitalization on the individual/society as described by Erstad, Kjällander & Järvelä [16]. These changes, towards an understanding of the digital world through digital competence and programming, are not only valid for Swedish education, but is according to Bocconi et al [17] a development seen globally.

According to a meta study by Lye & Koh [2] with data from 27 studies in different countries, programming and computing education is highlighted as something more than coding. This is related to *computational thinking* as described by Wing [1] and most studies lean on a knowledge view where students create when they program – which in turn consolidates what they have learned. STEAM has, according to Cervera et al [12] become a source of innovation for teaching and learning. Digital tools and systems are used to enhance student's interest and learning. In K12 STEAM education, programming has gained in relevance. In Sweden, digital competence and programming has been part of the curriculum since 2018 [6]. Programming in Mathematics focuses on algorithms/basics of programming while programming in the subject Technology is about understanding the role programming plays in our environment and skills to execute technological solutions. In Math, students are, according to the National Agency for Education [18] expected to train computational thinking in programming activities: *"...pupils should be given opportunities to develop knowledge in using digital tools and programming to explore problems and mathematical concepts, make calculations and to present and interpret data."*

This paper is part of a longitudinal project: *Programming Didactics. Learning and Assessing Programming in Primary Education* that has a purpose to investigate how students learn programming and how knowledge can be assessed in order to provide scientifically grounded programming didactic science as well as an assessment framework. Results in the project show that about 65% of grade 1-3 students claim that programming is fun. In grade 4-6 about 35% agree and in grade 7-9 only about 15% say the same. This is not uncommon, instead, research by Kong et al. (2018) shows that in general, students' interest drops after grade 3. It is not known why students' positive attitudes decline, but it might be related to teaching. This paper concentrates on teachers' didactic design and on how students learn programming. Empirical examples cast light on didactic designs that are appreciated by the students and on what they learn. Prior research on teaching and learning will be presented below.

II. Prior literature on teaching and learning programming

Prior research has been divided into three parts as these are found in the empirical material and since they are the most relevant for the results and the discussion: "digital competence", "collaboration" and "programming with visual characters".

Digital competence - A Swedish research review by Alexandersson & Limberg [20] shows that students' learning is changed by digitalization in four different ways: a) conditions for learning change; b) conditions for how students take responsibility and construct knowledge change; c) new conditions for the division of responsibilities and meaning-making arise, and d) the communicative structures of the school change. The aim is to make adults of tomorrow more digitally competent than the adults of today and a means to reach this goal is to teach computational thinking and programming in school. Prior research by Kjällander, Mannila, Åkerfeldt & Heintz [21] shows that teachers understand programming as something new and different, but for children it is just one of many new assignments, no stranger than for example multiplication. The same article shows that teachers struggle to gain digital competence simultaneously with their pupils. This phenomenon can be compared to "the twin imperatives for technology" similar to what Albion & Tondeur [11] described when teachers try to learn the same technology as their students – alongside with their students.

Collaboration - Prior research by Israel et al [22] shows three modes of collaboration: teacher-prompted; organized; and student-initiated collaboration. Collaboration with society is not uncommon. Instead, research by Kjällander, Åkerfeldt, Mannila & Parnes [23] illustrates how cooperation between education and society can have potential for learning programming when the formal and the informal meet – something that is visualized by education and Makerspace cooperation operations – where for example Arts play an important role. According to Facer [24], action with informal educational qualifications leads to the local community seeing the school as a place where they can contribute *and* get help. It can, according to Löfdahl, et al [25] affect people's and society's well-being and become a crucial meeting between society and its new generations Students who supported each other in programming activities have been studied by Cervera [12] showing that teachers mean that students who were mentoring were motivated when exploring through playing, investigating, observing, and cooperating. The use of robotics had positive contributions to autonomous work and the programming activities allowed students to engage in implementing algorithms, debugging and testing. Young students who were programming, both in independent learning and group learning in a study by Robertson & Howells [3] were motivated and enthusiastic to learn.

Programming with visual characters - Research by Robertson & Howells [3] shows that students designing computer games were positive to game programming: they linked and used their learning in new situations. Prior research on Scratch (which was used for learning basic programming concepts in a cross-curriculum setting) by Sáez López et al [26] showed that students understood computational concepts by using the program. Research by Cederqvist [27] indicates that understanding programmed technical solutions comprise a lot more than being able to write code. Students must learn programming concepts. While block programming, students who do not understand the code or the logic with the blocks, may use them as a jigsaw depending on their shapes. Instead, they must understand how blocks can be combined to control technical solutions. Research by for example Christensen, et al [4] and Kazakoff et al [5] suggests that younger children find it easier to learn programming by using visual or physical resources.

III. Theoretical perspectives: Designs for learning

This article is based on a view of interaction, meaning-making and learning as meaning making multimodal activities where modes such as sounds, images, gestures and colors are as important in communication and learning as speech and text. Within the framework of social semiotics as described by van Leeuwen [8] and multimodality by Kress [7], researchers have developed a design theoretic perspective presented by Selander [10] and Selander & Kress [9]. This perspective aims to investigate teachers' orchestration of multimodal resources in teaching – *design for learning* – and to understand students' learning – *design in learning* – where digitalization plays a crucial role. According to Selander [10] [28] learning is situated in social and cultural contexts: a design process of semiotic transformation/transduction and formation. Learning depends on *affordances* as originally described by Gibson [29] and developed by Kress [7] as a potential meaning. Kress [7] also present *prompt* as an actualized meaning in relation to individual interests. *Signs of learning*, as described by Selander [10] are formed and visualized, in digital or physical representations when students use their new knowledge in a new context. They can for example transform something they read into their own text or transduce something they read into another mode, such as a drawing. In school, students' multimodal communication and signs of learning are assessed formatively and summatively aligned research by Black & Wiliam [30] in teacher interventions, depending on the school’s and the, what as Selander would call [10]: environment´s recognition culture executed as assessment. A model to guide collection of empirical material, as well as to guide analysis of the same, has been designed: A Learning Design Sequence: LDS.

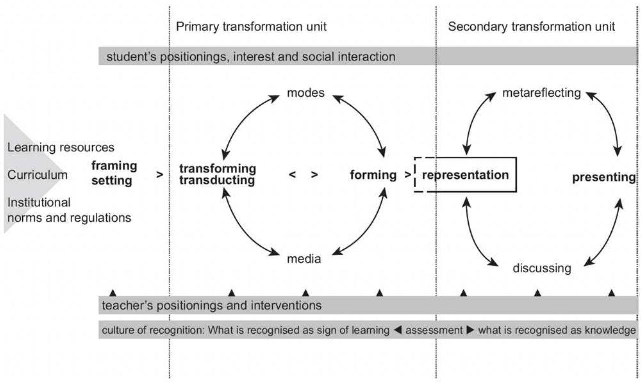

Figure 1: Learning Design Sequence, Selander [28].

An LDS is a series of lessons, from the teacher`s introduction of a subject area to the students´ presentation of their learning and finally the evaluation or/and the grading. An LDS can consist of everything from two lessons two days in a row to all lessons in a subject area during a whole semester.

IV. Aim and research questions

The aim of this paper is to find out how teachers can design programming activities so that students engage and learn. RQ: What affordances in the didactic design are appreciated by the students and what signs of learning in programming are visible? The longitudinal research project studied grades 1-9 during three years. This article reports data collected in middle school, with empirical examples from grades 4, 5 and 6.

V. Outlines of multimodal research methods

This study has a multimodal theoretical perspective, therefore multimodal research methods such as video documentation and collection of students’ representations such as texts and drawings have been made.

The purpose of the large longitudinal project (*Programming Didactics. Learning and Assessing Programming in Primary Education*[1]) is to investigate how students learn programming and how programming can be assessed in order to provide scientifically grounded programming didactic for primary education. The study will allow us to better understand how students learn programming and what good teaching practices are.). The project uses mixed methods. Two municipality run schools with approximately 1500 students in total are studied for three years. They are both 1:1-schools (all students have their own computer). Two classes in grade 1, two in grade 4, and two in grade 7 are followed for three years, thus empirical material from all compulsory school grades (1–9) are collected. The mixed methods project collects data using, for instance, questionnaires and video observations. Video observations in the classroom, collection of students' programs/films/games/images and questionnaires have been collected and analyzed with multimodal transcription schemes, where different modes have been divided into columns. The research project went through an ethical review by the Swedish Ethical Review Authority[2] and all ethical guidelines by The Swedish Research Council [31] have been thoroughly followed. Information meetings have been held and informed consents have been collected from all teachers, students and guardians. If students declined their participation, this overruled the parents' wish to participate.

This article presents video documentation and students digital and physical representations from one class (grade 4-6) with 30 pupils (aged 9-13 years) and 3 teachers. Three series of lessons or project areas, what Selander [10] would call LDS (Learning Design Sequences) have been followed in a class at one school. Two lessons at each grade have been video documented which sums up to approx. 240 mins. of film. Other lessons in the LDS have been observed, but not filmed, to understand the larger setting, but these have not been transcribed nor used in the article. One teacher interview per grade has been recorded which sums up to approx. 180 mins. We use multimodal analytical concepts to focus on communication in modes (gestures, images, sounds), as well as the actions that students make (codes, instructions) or their representations (robot’s movements, drawings) to depict their learning. This is described by Selander & Kress [9] as *signs of learning*. Student’s learning is consolidated into knowledge when they use their new knowledge in advanced ways in new situations. They can for example use an artistic

[1] Project number: Dnr MAW 2017.0096
[2] https://etikprovningsmyndigheten.se/

expression they just learned in Arts to present their new knowledge in for example Math. The empirical material consists of video documentation of classroom interaction, recorded focus group discussions with teachers and student´s representations such as texts, images and games. In grades 4 and 5, lessons in Technology were followed while Math lessons were followed in grade 6.

All films have been analyzed but only a few minutes of each lesson have been multimodally transcribed. C*ritical incidents,* a notion by Flanagan [32], have been selected from classroom interaction and *units of analysis*, as described by Matusov [33] have been transcribed. A design theoretical analysis, based on multimodal transcription, was made of the qualitative research material. The material was transformed into different columns for different modes such as drawings, speech, gestures, poetry, sounds, and screen activity. This paper presents a selection of representative excerpts from the transcriptions from each grade, followed by a short analysis by means of appropriately selected analytical concepts. This is a case study, which is limiting.

VI. RESULT AND ANALYSIS

Current results indicate that educational practice in middle school in Sweden, aligned with the curriculum, can afford teachers to design didactically so that their students learn to program. A few critical incidents [32] from each grade have been selected and are here multimodally illustrated and units of analysis [33] from video empirical classroom have been transcribed. *S* stands for student (if there are two students, they are named SA and SB), *T* stands for teacher, *V* stand for volunteer and *R* stands for researcher.

*A. **Grade 4***

In the beginning of middle school, in Grade 4, the importance of digital competence and the skill to program to be able to take part in the world and in democratic conversations is introduced by the teacher. The setting and the didactic design thus aim at everyday life and lifelong learning rather than school criteria and grades. A recurring didactic design is co-operation with the society aiming at an understanding of the role of programming in everyday life. A teacher invited a volunteer from a non-profit organization [3] to class to introduce programming with discussions about the human-technology relationship. Students' own experiences are highlighted and their concerns about artificial intelligence (AI) are discussed when a student asks the following:

S: Why is old AI so stupid? V: That's right. Artificial intelligence is getting smarter and smarter every day.

Students are then allowed to program a robot by giving step-by-step commands to the volunteer.

Next didactic design is to teach and then make students communicate in codes – with ones and zeros – with each other in notes.

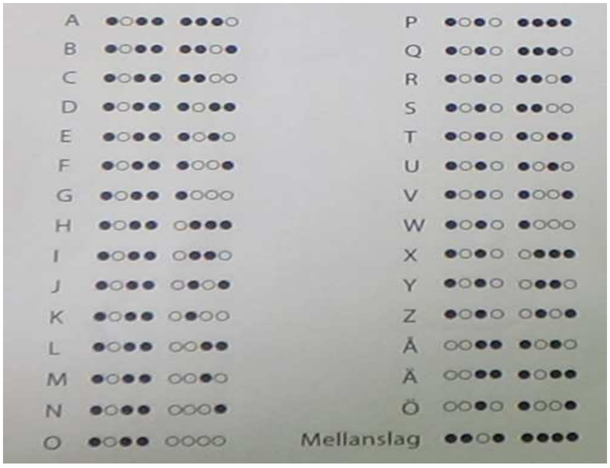

Figure 2. Chart handed out to the class with the assignment to write secret messages to a peer. The following information was given: Ones and zeros, or power on and off, is the language that computers speak. ASCII code is one way that we can translate letters and other characters into binary code. ... White dot is 1 (current on), black dot is 0 (current off).

Modes and media are here computers, a matrix along with paper and pen. Students´ representations are coded letters written to each other. Their signs of learning are visible in both how they decode and understand each other's messages. Information is here transduced from symbols to text.

In the second transformation unit in the LDS an aesthetic affordance (or prompt even) was that students had to meta reflect and write about what they have learned. The teacher included Arts in the didactic design and told them to write poems about their understanding of the impact of computers and digital media in society. Here is a poem written by two grade 4 students:

Yellow and blue we love coding.
It stands for the world and ones and zeros.
The future of the world depends on coding.
The center of the world is ones and zeros.
In other words, coding.
All of Sweden and the world loves coding.
Yeah yeah yeah!

The objective of this LDS was to introduce students to computational thinking by getting acquaintance with notions and principles behind the technology. This was made through deconstructing algorithms as well as to make them understand that digital communication at the bottom line consists of 1/0 or on/off – something that the poem above seems to illustrate.

*B. **Grade 5***

In grade 5 students are designing their own games in Tynker. In the primary transformation unit, they program together at the whiteboard trying different instructions. The didactic

[3] Kodcentrum - a non-profit association that, with the help of non-profit forces in schools, authorities, municipalities and industry, introduces children to programming and digital creation for free. https://kodcentrum.se/

design is about identifying mistakes and debugging them. In this LDS the teacher motivates students to meta reflect on their own learning and to cooperate with each other. The teacher explained that she is new to programming as well and that they must support each other:

T is standing in front of class by the whiteboard, gazing at the class: I have done a few courses in programming, but I have not received as much training as you have received! T shakes her head. You've trained more than I have. And when you run into problems and need to solve them, we have to work together to move along and to move forward. And what kind of ability are you then training? What are you thinking, A? T turns to a student. A: Ability to cooperate. T: Yes. T smiles and nods.

This narrative didactic design seems to be appreciated as an affordance by the students who are cooperating and helping one another, although working with their own game. Later, the teacher asks the class what skills are important in problem solving and programming. One student answers “creativity”, thereafter the teacher asks the student to explain what it can be:

S: You are creative when you come back and when you create. T nods: Yes, you could say that, new things, and to mix things… and things that maybe at first... maybe you didn't think they would fit in! Think a little outside the frame! T is gesticulating a square frame in the air.

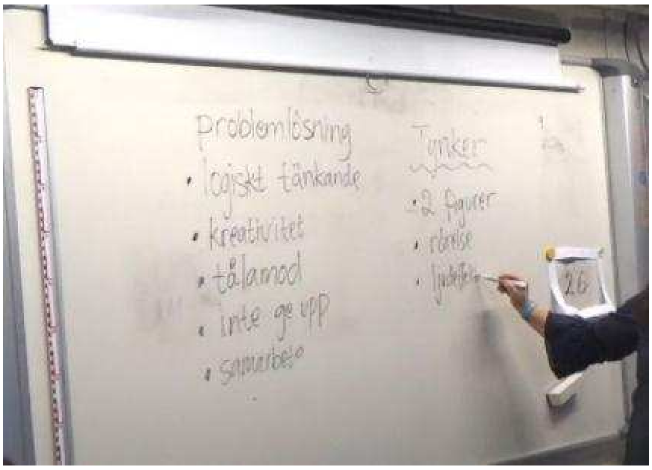

Figure 3: White board notes about key concepts on programming/problem solving: logical thinking (referred to as computational thinking), creativity, patience, grit and cooperation. The assignment (what the Tynker game must include): 2 figures, movement and sound effect.

By the end of the primary transformation unit, the students present their games to classmates to get peer feedback. The game design presented in this example illustrates a figure that must avoid burning comets in order to survive, before a countdown from 5-1. The student (A) was keen to design his own numbers and images or transform images from Tynker/Internet by manipulating them.

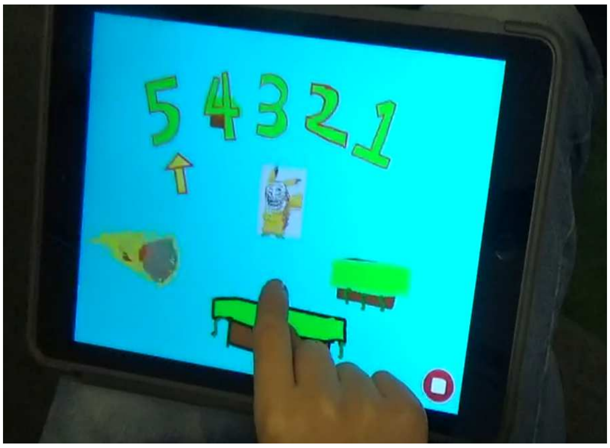

Figure 4. Multimodal game design in Tynker makes use of Arts (cf. STEAM).

R: What happens if one manages to jump over all the comets, then? SA: I haven't figured that out yet. R: No. Then maybe you win? SA: Yes! SB. I think it looks very good. R: Yes, really... Very advanced. That's great! SA: It's just that when you've lost once, you don't really know when... SB: Maybe you should... There is a block that makes you stop everything. SB is pointing at the screen. R: Yes, that's right. What happens when you stop everything? SB: Yes, then... Everything just stops. SB: Yes, that's right. SA: That's what I did with my game. I got it. Here. SA points at the screen. SB: Well. A: But it, kind of, doesn't work. SA shake his head.

Figure 5. Meta reflection and peer assessment.

In this example, the different Tynker blocks are considered affordances offering the student to form individual game designs. Signs of learning are visible in the conversation between the students where student B:s earlier experience is shared and student A transforms and forms his game accordingly. In the second transformation unit all students present their games, explaining how they worked, what they had trouble with designing, how they designed the objects artistically and how they came to solutions while transforming and forming the two figures aesthetically, the movements and the sound effects. This kind of metareflective peer assessment didactic design, visualizes signs of learning vividly. Students explained what they did not manage and then how they transformed and formed the assignment in

Tynker. This didactic design relates to computational thinking by problem solving, highlighting failures as examples to learn from and by thinking creatively, aesthetically and logically together in groups.

C. *Grade 6*

In the setting of this LDS the teacher prompts the students to recall the different programming apps they have used during grade 4 and 5. Tynker, Scratch, Kodcentrum, Bluebot, Lightbot and other apps have been used. Scratch is then chosen by the class to design a useful digital tool that can be used in school.

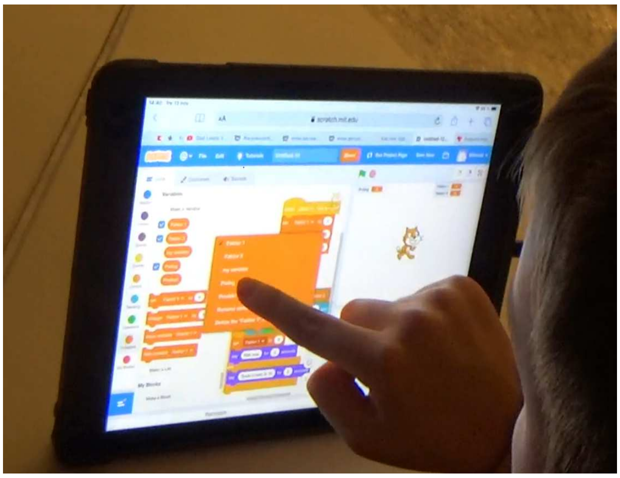

Figure 5. Block programming in Scratch.

Student A is designing a calculator for multiplication with images of his own choice. Different blocks are set in the right positions and the students cooperate, although designing their own tools.

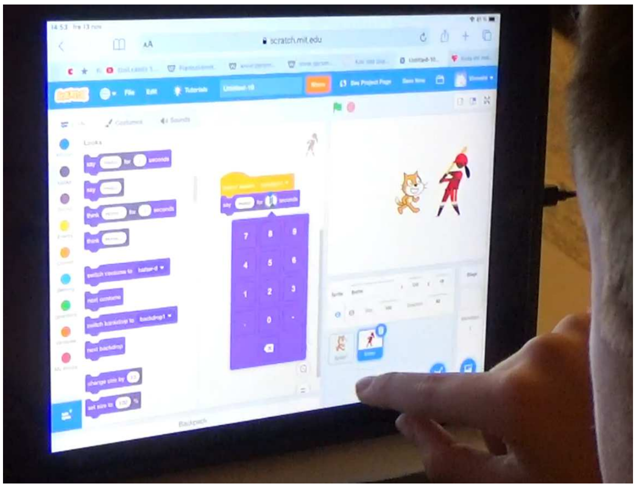

Figure 5. Designing a calculator in Scratch.

SB: SA? On this one: this dot, the one, is it something important or can you take that one instead? SB points at the screen. SA: No, it's just that, it has to be times. SB: Aha, there should be multiplication signs? SA: Yes.
…

Figure 6. Calculator question: What is 7 times 8?

S: Now I'm going to apply this sequence. Yes, you have to put in... this... S places number 7 and number 8 in the right places and writes 56. S: So! Yes, it's working! R: Good! S: It wasn't that complicated! There are only two short... R: Well, it's pretty complicated. S nods: Yes. Especially when you're going on like… S is moving his hands and talking with a funny voice. when you're coding with: Say... blablablablablabla. (What-if-statements). Yes, then you need to know all the codes, like (commands), by heart!

Figure 7. Calculator answer: It is 56.

Signs of learning can here be visualized in how students know how important it is to choose the right punctuation, number or block to make the object/tool do what is intended. What if-statements that execute a set of instructions if a special condition is met (and another action if the special condition is not met) are thus yet not learned.

In the end of middle school, and in the second transformation unit of this LDS, programming is evaluated in class:

T: If you look back, like, what do you think you've learned so far that has to do with programming? What are you taking with you, where do you stand now? What do you say, S? T turns to S. S: Block programming! T nods: Mm. And it's also a little that division we have made here at the school: that the middle school is about block programming.

Students' multimodal communication is assessed formatively and summatively [30] during the LDS due to the school's recognition culture that school constitutes [10] and that teachers execute as meta reflection or assessment. In this final LDS the students and the teacher agree that they have learned what they were supposed to during middle school in the programming lessons in Technology and Math: block programming.

VII. Discussion

This part of the article will discuss how teachers design programming activities so that students in grade 4, 5 and 6 can engage and learn. The research question concerned the affordances in the teacher´s didactic design that students appreciated, and what signs of learning in programming that were visible in the empirical material.

The LDS in grade 4 (Math) and 6 (Technology) focuses on digital competence and students' ability to understand and program in their digitalized everyday lives, related to Wing´s [1] computational thinking's parts about familiarizing with the concepts and principles underlying the technology and problem solving. The didactic design relates to democracy compared to Karlsohn [13], Hernwall [15] and Samuelsson [14] who all mean that digital competence is a prerequisite for participation and influence in society. The aims of the lessons in the studied LDS:s are all about designing and creating something useful (such as a calculator). The Government [6] states six changes, that society must change to meet the new demands, which in short is: 1) introduce programming in school, 2) strengthen critical evaluation of sources, 3) problem solving creatively, 4) work with digital texts, 5) use digital systems 6) understand the impact of digitalization on the individual/society. These changes sum up to education "*towards an understanding of the digital world through programming*" [16]. In grade 5 (Math), on the other hand, digital skills and abilities are practiced comprising computational thinking by means of reversing/deconstructing algorithms and commands along with the ability to think creatively, "out of the box" together with highlights on mistakes, as examples to learn from. Aligned with prior research by Lye & Koh [2], students engage in computational thinking while creating digital representations. Their learning is consolidated into knowledge when they use their new knowledge (for example using the right punctuation) in advanced ways (as a calculating instruction) in new situations (such as an own programmed calculator): signs of learning. Both aims correspond to the curriculum's writings that students should develop knowledge in using programming to explore problems and mathematical concepts and make calculations as well as to present and interpret data aligned the curriculum [18].

The empirical material in this research project present LDS:s where teachers design for learning in cooperation with the society. In this paper society is represented by a non-profit coding organization: Kodcentrum, whose volunteers' narratives, assignments and activities are appreciated as affordances by the students. The student become interested and engaged but also skeptical and anxious about how digitalization can change society. In a discussion about instructions and robots in class, thoughts about how smarter AI can gain power over human, worries recognized in research by Karlsohn [13] and Hernwall [15]. Students' signs of learning are multimodal in for example their speech, images and texts. These conversations are, according to the curriculum [18] important and are here understood as a "crucial meeting between society and its new generations" as mentioned by Löfdahl et al. [25].

The teachers' outspoken aim in the setting of an LDS (whether aimed at digital competence/understanding of the digital world or aimed at digital skills) and narratives seems to be appreciated as potential meanings – affordances – by the students. Teachers, who *design for learning,* are becoming digitally competent simultaneously as their students, while intervening in students' *design in learning*. The so called "twin imperatives for technology" as described by Albion & Tondeur [11] can make teachers and students engage in genuine cooperation. This is explained by the teacher in grade 5, who says the students have trained programming more than she has and encourages everyone to help one another – no matter if she is a child or an adult. Thus, the communicative structures are changed in the digital learning environment, just as Alexandersson & Limberg [20] foresaw. Students are prompted to cooperate with teachers and with each other to solve programming tasks. All Israel et al's [22] three kinds of collaboration are represented in these LDS:s. In contrast to Robertson & Howell's [3] research indicating that students working together as well as individually are as motivated and enthusiastic to learn programming, students in this study collaborate enthusiastically at all times (when allowed).

A majority of LDS:s make use of visual resources (such as images, student's own drawings or blocks in Scratch). This is, both in this study and in prior research by Christensen, et al [4] and Kazakoff, et al [5], understood as an easier, or more attractive, way to learn programming. Students seem to appreciate affordances in the aesthetic/visual programming environment. Equally, in the didactic design of representing new knowledge in a poem, aesthetics seems to be appreciated as an affordance. Students´ signs of learning are visualized in aesthetic expressions. Students transform their experiences (matrixes of code) and form a representation (poem) where their new knowledge (about digital communication with ones and zeros) is presented in the mode of art, as lyrics: "*The future of the world depends on coding. The center of the world is ones and zeros.".* This is an example of how Science, Technology, Engineering, Arts and Mathematics can merge within the frames of programming and digital competence.

VIII. Findings of value and interest to computing educators

This article gives insight to *design for learning* and *design in learning* in middle school. Especially three aspects will be presented as important for teaching and learning programming in middle school: collaborative work, computational thinking, multimodal/aesthetic expressions in STEAM.

To begin with, our study, even though it is small-scale, indicates that collaboration is a successful didactic design for programming in school. Even if the "twin imperatives of technology" feels awkward to teachers, students appreciate it as an affordance to help and collaborate with each other and with the teacher. In their cooperative work, such as meta reflecting on their learning process and on their new knowledge and in peer assessment of each other's products their signs of learning are visible. Collaboration also occurs with the society.

Secondly, computational thinking is prevalent in all LDS:s. Both digital skills today in the classroom, and digital competence for the future to be able to participate in a digitalized democratic society, are practiced and often met. Students´ signs of learning are seen in their discussions and in their poetry.

Finally, Art, visual modes and multimodal narratives are appreciated as affordances, or even prompts, by the students. Block programming is prominent in middle school. Students' signs of learning are visible while they form, transform and transduce their new knowledge in programming by means of aesthetics in advanced ways in new situations - which are often framed by STEAM.

IX. Acknowledgements

We want to thank students and teachers who participated in this research. The project is funded by Marcus and Amalia Wallenberg Foundation.

X. References

[1] Wing J. Computational thinking. *Communications of the ACM*, 49(3), 2006, pp. 33–36.

[2] Lye, S.Y., & Koh, J.H.L. Review on teaching and learning of computational thinking through programming: What is next for K-12? *Computers in Human Behavior* 41, 2014, pp. 51–61.

[3] Robertson, J., & Howells, C. Computer game design: Opportunities for successful learning. *Computers & Education,* 50 (2), 2008, pp 559-578.

[4] Christensen, D.J., Fogh, R., & Lund, H.H. Playte, a tangible interface for engaging human-robot interaction. *Robot and Human Interactive Communication*, RO-MAN: The 23rd IEEE International Symposium, 2014, 56-62.

[5] Kazakoff, E.R., Sullivan, A., & Bers, M.U. The Effect of a Classroom-Based Intensive Robotics and Programming Workshop on Sequencing Ability in Early Childhood. *Early Childhood Education Journal*, 41 (4), 2013, pp. 245-255.

[6] Government. *Nationell Digitaliseringsstrategi för Skolväsendet*; Bilaga till regeringsbeslut I; Swedish Government: Stockholm, Sweden 2017, Volume 1.

[7] Kress, G. *Multimodality: A social semiotic approach to contemporary communication*. London: Routledge, 2010.

[8] van Leeuwen, T. *Introducing social semiotics.* London: Routledge, 2005.

[9] Selander, S. & Kress, G. *Design för lärande: Ett multimodalt perspektiv*. Stockholm: Norstedts Akademiska Förlag, 2010.

[10] Selander, S. Didaktisk Design. In: *Didaktisk design i digital miljö – nya möjligheter för lärande.* Selander, S. & Svärdemo-Åberg, E. (Eds.) Stockholm: Liber, 2009.

[11] Albion, P. R., & Tondeur, J. Information and Communication Technology and Education: Meaningful Change Through Teacher Agency. *In Second Handbook of Information Technology in Primary and Secondary Education*, Springer, 2018, pp. 381-396.

[12] Cervera, N.; Diago, P.D.; Orcos, L.; Yáñez, D.F. The Acquisition of Computational Thinking through Mentoring: An Exploratory Study. *Educ. Sci.* 2020, *10*, 202.

[13] Karlsohn, T. Samhället och teknikskiftet. I T. Karlsohn (Red.), *Samhälle, teknik och lärande.* Stockholm: Carlsson, 2009.

[14] Samuelsson, U. *Digital (o) jämlikhet? IKT-användning i skolan och elevers tekniska kapital* (Doctoral dissertation, School of Education and Communication), 2014.

[15] Hernwall, P. 'We have to be professional'–Swedish preschool teachers' conceptualisation of digital media. *Nordic journal of digital literacy*, *11*(1), 2016, pp. 5-23.

[16] Erstad, O., Kjällander, S., & Järvelä, S. Facing the challenges of 'digital competence' a Nordic agenda for curriculum development for the 21st century. *Nordic Journal of Digital Literacy*, 2021, 16(2), 77-87.

[17] Bocconi, S., Chioccariello, A., Kampylis, P., Dagienė, V., Wastiau, P., Engelhardt, K., Earp, J., Horvath, M.A., Jasutė, E., Malagoli, C., Masiulionytė-Dagienė, V. and Stupurienė, G., Reviewing Computational Thinking in Compulsory Education, Inamorato Dos Santos, A., Cachia, R., Giannoutsou, N. and Punie, Y. editor(s), *Publications Office of the European Union*, Luxembourg, 2022, ISBN 978-92-76-47208-7, doi:10.2760/126955, JRC128347.

[18] *Curriculum for the compulsory school, preschool class and school-age educare.* (revised 2018). Skolverket: Stockholm, Sweden, 2019. https://www.skolverket.se/publikationer?id=3984 ISBN : 978-913832734-0 (accessed on 15 February 2021).

[19] Kong, S. C., Chiu, M. M., & Lai, M. A study of primary school students' interest, collaboration attitude, and programming empowerment in computational thinking education. *Computers &education, 127, 2018, pp. 178-189.*

[20] Alexandersson, M., & Limberg, L. Changing conditions for information use and learning in Swedish schools: A synthesis of research. *Human IT: Journal for Information Technology Studies as a Human Science*, 2012, *11*(2).

[21] Kjällander, S., Mannila, L., Åkerfeldt, A., & Heintz, F. Elementary Students' First Approach to Computational Thinking and Programming. *Education Sciences*, 2021, 11(2), 80.

[22] Israel, M., Pearson, J.N., Tapia, T., Wherfel, Q.M., Reese, G. Supporting all learners in school-wide computational thinking: A cross-case qualitative analysis. *Comput. Educ.* 2015, *82*, 263–279.

[23] Kjällander, S., Åkerfeldt, A., Mannila, L., & Parnes, P. Makerspaces across settings: Didactic design for programming in formal and informal teacher education in the Nordic countries. *Journal of Digital Learning in Teacher Education*, 2018, 34(1), 18-30.

[24] Facer, K. Taking the 21st century seriously: young people, education and socio-technical futures, *Oxford Review of Education*, 38:1, 2012, 97-113.

[25] Löfdahl, A., Hägglund, S., & Thelander, N. *Förskolans demokratifostran: I ett föränderligt samhälle*. Stockholm: Gleerups, 2022.

[26] Sáez-López, J.M.; Román-González, M.; Vázquez-Cano, E. Visual programming languages integrated across the curriculum in elementary school: A two year case study using "Scratch" in five schools. *Comput. Educ.* 2016, 97, pp. 129–141.

[27] Cederqvist, A. M. *Seeing the parts, understanding the whole-A technology education perspective on teaching and learning in processes of analysing and designing programmed technological solutions*, Gothenburg studies in educational sciences, Doctoral thesis, 2021.

[28] Selander, S. *Didaktiken efter Vygotskij: design för lärande*. Liber, 2017.

[29] Gibson, J.J. *The Ecological Approach to Visual Perception*; Houghton Mifflin: Boston, MA, USA, 1979.

[30] Black, P., & Wiliam, D. Developing the theory of formative assessment. *Educational Assessment, Evaluation and Accountability,* 2009, 21(1), 5-31.

[31] The Swedish Research Council. *God forskningssed.* Stockholm, 2017.

[32] Flanagan, J. C. The Critical Incident Technique. *Psychological Bulletin*. 1954, 51 (4), p. 327-359.

[33] Matusov, E. In search of the appropriate unit of analysis for sociocultural research. *Culture & Psychology*, 13.3: 2007, pp. 307-333.