The Scope of Autonomy Model – Development of Teaching Materials for Computational Thinking in Primary School

(article starts on next page)

The Scope of Autonomy Model – Development of Teaching Materials for Computational Thinking in Primary School

Niklas Carlborg
RISE Interactive
Gothenburg, Sweden
niklas.carlborg@ri.se

Markus Tyrén
RISE Interactive
Gothenburg, Sweden
markus.tyren@ri.se

Carl Heath
RISE Interactive
Gothenburg, Sweden
carl.heath@ri.se

Eva Eriksson
Aarhus University &
Chalmers University of
Technology
evae@cc.au.dk

ABSTRACT

During the 21st century there has been an increasing interest in the field of computational thinking as a consequence of the ever faster technical development. However, educating future generations in programming and computational thinking is not trivial. Many different platforms and teaching approaches can be used for this purpose. Inspired by the UK initiative with BBC micro:bit, this paper strives to identify what may be important to consider when designing teaching materials with the micro:bit for training Swedish primary school pupils' computational thinking skills relating to mathematical and technical school subjects. This has been investigated in an iterative process, by conducting 21 workshops with the goal to support primary school teachers in developing micro:bit teaching materials. The contribution of this paper is the Scope of autonomy model, which is based on the relation between pupils learning potential, their risk of feeling overwhelmed and the amount of choices provided in exercises. The model aim to support teachers in developing material for teaching programming and computational thinking in accordance with the new curriculum.

Author Keywords

Computational thinking; Autonomy; Micro:bit; Education; Teaching.

CCS CONCEPTS

• Human-centered computing → Interaction design; • Applied computing → Education;

INTRODUCTION

In a society with accelerating technical development, there are many challenges in designing education that can prepare next generations for the future. With an ever faster technical development it becomes less relevant to teach specific skills that might become obsolete in a near future, instead it becomes more important to teach skills that enable new

generations to swiftly adapt to the changes and technologies that emerge. The rapid technology development calls for a 'fluency' approach, rather than the traditional 'skill-based' approach [1; 2], and there is a central challenge in learning how to work with technology and design through iterative, reflective and flexible approaches to learning [3].

Sweden is set to introduce programming in primary and secondary curriculum in 2018 [4]. The main focus of the changes in the school curriculum is to enhance and emphasize the school's role in strengthening the pupils digital competences, varying from teaching stepwise instructions in the early year groups to fully encompass programming in later year groups. There has been little research done on how the teacher's technological skills and attitudes towards technology will affect such a transition nor on what school resources will be required. The current research has focused on impediments that arise regarding the shift in mindset that is required from teachers in a more explorative teaching setting, rather than the more traditional goal oriented approach [3].

In the UK the transition to include programming in education has already begun. As part of the Make It Digital initiative in 2015, BBC has together with Microsoft, Samsung and other partners, developed the micro:bit for use in computer education [12]. Every pupil aged 11-12 in the UK was given one of these microcontrollers that can be programmed and customized. It aims to inspire young people to be creative in the digital world, developing core skills in STEM subjects and produce a new generation of inventors and makers.

Inspired by the UK, a project was initiated to explore how micro:bit can be adapted to aid in the Swedish curriculum transition. This paper reports from a study in a collaboration between university researchers and a national research institute aiming to investigate what is important to consider when designing teaching materials with the micro:bit for training primary school pupils computational thinking skills. The contribution of this paper is the Scope of autonomy model for developing teaching materials with a technological platform, such as the micro:bit, for training computational thinking in primary schools. The results is based on non exhaustive empirical design research and is limited to the micro:bit platform and the Swedish school context.

BACKGROUND

Hattie and Donoghue propose a model of learning [6] based on three inputs and outputs: skill, will and thrill. The model mentions the importance of defining the success criteria to the learner and that there are three phases of learning: surface, deep and transfer. Surface and deep are also each divided into an acquiring and a consolidation phase. The model suggests that some learning strategies are more effective than others but that this is dependent on the learning phase. Further, it is argued that learning strategies should be embedded into subject content rather than be taught separately out of context. Transfer is shown to be highly effective for learning, especially in looking at similarities and differences between different contexts and situations. It is suggested that transfer requires the previous phases to be passed in a linear fashion. However, the authors also point out that this is an assumption and that more research is needed on the order of phases impact learning.

Self-Determination Theory (SDT) is a macro theory of human motivation, and claim that motivation has more dimensions than simply the strength amount of motivation [8]. Central to SDT is that psychological well-being and performance are predictable on three basic needs: autonomy, competence, and relatedness. To the extent of which these three needs are satisfied or thwarted by the context an individual's differences are changed in two ways: causality orientations and aspirations. Causality orientations relates to three ways of orienting oneself in relation to regulating one's behaviors. These orientations are: autonomy, which is acting out of interest; controlled, which is focused on rewards, approval and gains; and impersonal or amotivated, which is an anxious relation to competence. According to SDT all individuals have degrees of all three orientations. These are suggested to be influenced by an individual's surroundings support for the three basic needs (autonomy, competence, and relatedness) and have been shown to correlate with a person's psychological acquired by an individual to compensate for thwarted basic psychological needs (autonomy, competence, and relatedness) over time. These goals are either intrinsic aspirations or extrinsic aspirations. Intrinsic aspirations are for instance affiliation, generativity, and personal development. Extrinsic affiliations include goals such as fame, wealth and attractiveness [8]. There are factors that can catalyze or undermine intrinsic motivation. Tangible rewards, threats, deadlines, directives, competition pressure and negative performance feedback undermines intrinsic motivation. On the other hand, choice and opportunities for self-direction, as well as positive performance feedback has been shown to enhance intrinsic motivation [7]. Behaviors that are not intrinsically interesting to a person will require extrinsic motivation to be adopted. To make an extrinsic motivation more self-determined is the process of internalization and integration. This is done when a learner truly understands the values of an activity, identifies with it and incorporates it with their sense of self. This is suggested by addressing the basic psychological need of

relatedness, by having the behavior valued by significant others to whom they would like to feel connected. Therefore, it is important to provide a safe comforting environment where the learners feels that they can trust the facilitators. To further support internalization and integration it is argued that the need for competence has to be supported through challenges where the pupil feel that they have the competence to succeed. To support internalization and integration to the extent that the regulation becomes autonomous however, the basic psychological need of autonomy has to be supported by the environment as well. This can be done in such a way that makes the pupil feel free and agentic to explore new ideas and exercise new skills [7].

During the 21st century there has been an increasing interest in the field of computational thinking (CT) [9]. Although the concept of CT being important in education is not new, as early as the 1960's there were those advocating teaching programming to college pupils. Most notably was Seymour Papert's MIT work with the program LOGO in the 80's, as this was aimed at K-12 education [5]. This is based on constructivism, that knowledge is a structure built in the mind of the learner rather than something prepackaged ready to be absorbed from the teacher. In Papert's constructionism however also adds that the learner constructs this knowledge while consciously creating some public entity [5].

Although the concept of CT in education stems from 1960's, most of the recent work regarding CT has been developing tools and definitions, less focus has been put on the assessment [10]. However, one example is the MIT model. The MIT model is a CT framework split into three categories: computational concepts; computational practices; and computational perspectives [11]. In order to assess the level of CT, there are three strategies that can assist: 1) Artifact-based interviews about projects and practices, using examples to guide the conversation forward, 2) Design scenarios for the learners to engage in from four different angles: critiquing; extending; debugging; and remixing, 3) Documentation for reflection on creations and ideas.

As digital fabrication technologies makes increasing impact on STEM subjects in primary and secondary school, the teacher's role to handle these new learning processes of both technology and design has been largely overlooked. Smith et al [3] have identified four impediments with the teacher role in focus for integrating technology to support education. Firstly, there is a tension between the goal-oriented traditional school and the explorative nature of digital fabrication materials and practices. Secondly, a change in mindset is needed, where fabrication materials as reflection tools rather than just outcomes of a design process can be a contributing factor to be able to integrate digital fabrication in the classroom. Closely tied is the need for a design language, a common ground of understanding between teachers and pupils to express ideas and qualities regarding design. Fourthly, teachers have to get accustomed to having

less authority and less control due to not mastering all the techniques that are taught so there is a need for the teacher in a facilitator role.

In this paper, inspired by the listed learning approaches, we aim to identify what factors are important for successful integration of computational thinking in schools, and that can help the teachers meet the programming requirements of the new curriculum changes. More specifically, we aim to support teachers in designing exercises with the micro:bit platform accommodated to the children's different knowledge levels.

RESEARCH METHOD

The project was based on a Human-Centered design approach in order to investigate what factors are important concerning using micro:bit for teaching computational thinking in Swedish schools. Initially, we gathered information about how teachers and pupils perceive the introduction of programming in school through related literature. Additionally we familiarized us with the platform, micro:bit, and iteratively developed exercises to be practiced during a series of workshops in schools. Based on this, we created a script that was used consistently together with all workshops, see example in Table 1.

Table 1: Example of a workshop script

Workshop	Nr 1
Description	Analog first workshop in a series of four. Familiarizing with sequence and loop.
Time	Feb 23, at 9.40-11.00
Location	The school
Participants	17 students, 2 facilitators, 1 observing teacher
Age	4th graders
Aims	Test if analog workshop works as introduction to programming for complete beginners. Test a physical game as method for introduce sequencing Test if a physical game can work as a common experience to relate back to in future more detailed workshops See how fluent the transitions between different sections and workshops are, is there a common theme?
Methods	Exit tickets from students, and video observations of the facilitators
Insights	Teacher better execute the code Students like to present unique solutions

Insights from past workshops were incorporated the iterative development of new material. At a later stage, we created personas from the gathered and used them in a journey map

in an attempt to extract even more insights. Over a period of one year, we performed 21 workshops in school environments and educational conferences in different parts of Sweden. The workshops were led by two of the authors. The participating schools and teachers were recruited from the extended network of a national educational Maker-project in which the project is a partner. While field-testing the teaching formats and exercises, feedback from the testers and the process was collected for later evaluation. During the workshops, most of the data gathering took place almost exclusively through exit tickets, and additionally some oral feedback. Evaluation and analysis of data were carried out at the end of each prototyping cycle to assess in what extent the implementation was done as design implied. Insights were extracted using an inductive approach and used to prepare for the next cycle. In our final evaluation phase, all data was evaluated with affinity clustering. The results from the project work is the scope of autonomy model presented later in this paper.

The workshops

In order to identify what factors are important for teaching computational thinking with micro:bit in Swedish schools, a series of workshops were initiated, see Table 2 for overview. All the workshops were performed by the same two facilitators, and lasted for about one hour.

Table 2: Overview of the 21 workshops

Location	Age	Participants
Teacher conference	Adult	100 Swedish teachers
School	8-10 y	30 pupils, 1 teacher, a few parents
Lab	8-10 y	30 pupils, 1 teacher, a few parents
Lab	9th grade	2 pupils
School	4th-5th grade	20 pupils, 1 observing teacher
3x the same School	4th-5th grade, Adult	30 pupils, 1 observing teacher. Teacher workshop 15 participants
6x the same School	4th-6th grade	15 pupils
School	Mixed	30 pupils, 1 observing teacher
BETT show London	Adult	100 Swedish teachers
Lab	7-13 y	23 pupils
4x the same School	4th grade	17 pupils, 1 observing teacher

The exercises typically varied from analogue programming games to more advanced levels covering the basic concepts of programming: algorithms, loops, randomness, logic, variables, and finally debugging. The platform was BBC micro:bit programmable microcontroller, and was provided by the workshop organizers. The micro:bit was used together with the Microsoft MakeCode micro:bit editor, which is a free to use online and is a JavaScript/Blocks editor for programming the micro:bit. This means that it runs in the web browser and hence is cross platform compatible, both on different web browsers but also across different operating systems, such as OSX, Windows, iOS and Android. This also implies that an internet connection is required for using the editor.

The digital workshops included one micro:bit per pair of pupils, and the exercise was introduced by a co-coding approach. Co-coding is when the teacher stand in front of the class, screen sharing their computer screen on a projector and solve programming exercises in dialogue with the class. This form is a useful hybrid between pure presentations and having pupils solve exercises on their own. Presentations were considered beneficial for introducing new knowledge, but it was undesirable to put the pupils in a rather passive seat. On the contrary, working individually with exercises, was considered to be more active but not ideal for introducing novel information. Co-coding hence evolved as a middle path between these two approaches, see Figure 2.

Traditional Lecture	Group work
Teacher talking in plenum	Student in pairs, solving tasks
+ Introducing new knowledge	+ Active students
- Passive students	- Introducing new knowledge

Co-Coding
Teacher stands in front of class, sharing computer screen on projector to solve programming exercises in a dialogue with the class
+ Partly activates students
+ Introduce new knowledge
+ Quicker feedback supports the teacher to adjust level of difficulty

Figure 2: Co-coding as a combination of lectures and student work

Example of exercise with micro:bit: Rock Paper scissors
Rock Paper Scissors is a game that many people are familiar with, and this exercise combines several key programming elements. The program starts by shaking the micro:bit, and a number (0-2) is saved on variable "weapon". The code then proceeds by checking the number stored on the variable to match it with an image to be shown, see Figure 3. This is an engaging exercise as the pupils can test it out with their friends and compete when they are done. Also, there are several ways to continue building upon this program, e.g. by

implementing the game to work via radio or add code to keep track of the score. To provide extra feedback from the program, *pause* and *clear screen* can be added.
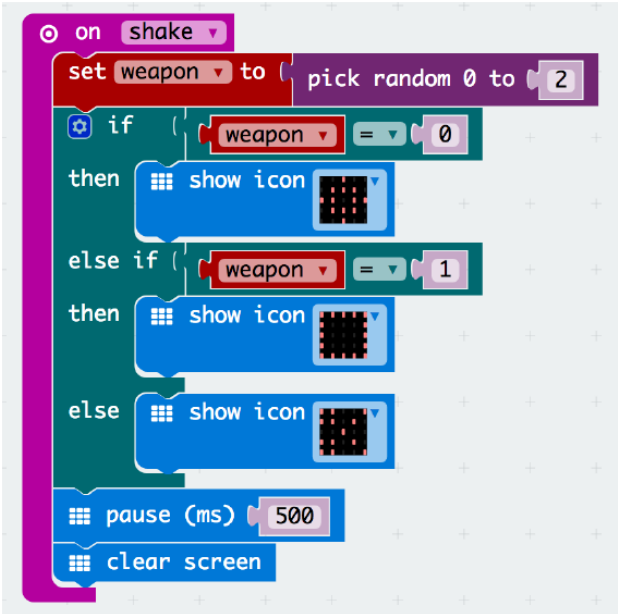

Figure 3: Rock Papers Scissors exercise

SCOPE OF AUTONOMY MODEL
The scope of autonomy model aim to explain observed behaviors and phenomena regarding primary school pupils encounter with programming the micro:bit. The model consists of five levels of autonomy, and is intended to be used as a tool when creating exercises, to help teachers bring awareness to the amount of choice expected of pupils within exercises.

The scope of autonomy model illustrate and bring awareness to the distribution of autonomy between pupils and teachers in relation to single given micro:bit exercise. The model is based on the premise that completing an exercise involves
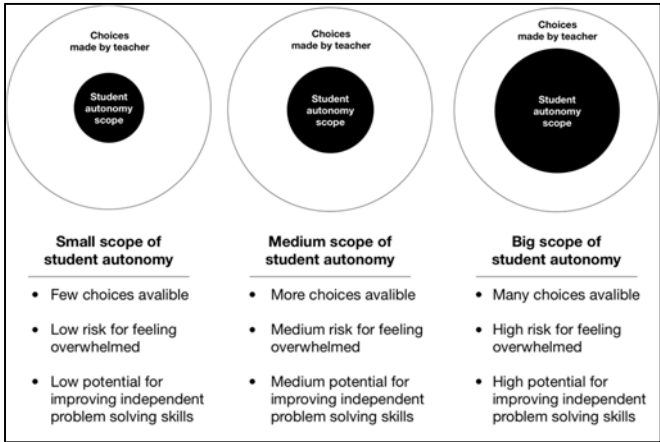

Figure 4: Scope of autonomy model

Customization

Students are allowed to customize small predefined parts of a fix solution. For instance choosing the text or image shown on a screen.

Solution procedure

Students are provided with a predefined design they are supposed to create, but they are free to choose the order in which they want to create it.

Design

Students are given a fix assignment with a predefined set of blocks to use, but they are free to create any design that satisfies the assignment.

Block selection

Students are given a fix assignment, but they are allowed to freely choose the type of blocks, and or the number of blocks, to use themselves.

Assignment

Students are allowed to choose their assignment themselves, which in autonomy is eqivalent to allowing them to pursue their own ideas.
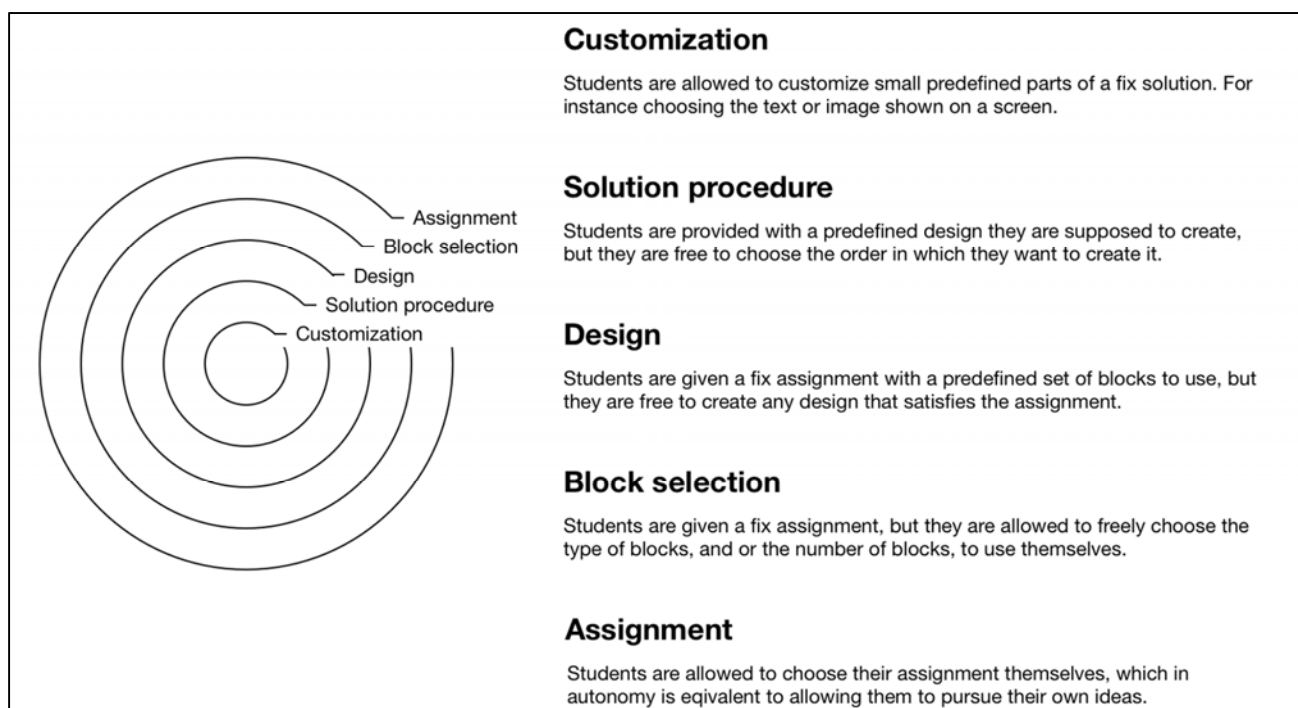
Figure 5: Each level represent a level of autonomy

making a set of choices. The dark area in the center of the model in Figure 4 represents the choices made available to the pupil, this is called the pupils scope of autonomy. The area surrounding it represents the choices made by the teacher. A larger pupil scope of autonomy hence implies fewer choices to be made by the teacher.

The model suggests that the larger the scope of autonomy becomes, the higher the pupil runs a risk of feeling overwhelmed. The model also suggests that the larger the scope of autonomy becomes, the higher the potential is for the pupil to improve their independent problem solving skills. Hence, there is a balancing act in the creation of an exercise, in that it provides the pupils with enough choices to develop their independent problem solving skills, yet without exposing them to too many choices so that they feel overwhelmed. The model does not make any claims on how to determine what the appropriate level of autonomy is for any pupil.

Five levels of autonomy were identified for working with the micro:bit. These are presented in a radial fashion to be compatible with the scope of autonomy model, see Figure 5. Any micro:bit exercise can be mapped as a scope of autonomy disc onto this model. The more of these levels that are encompassed by an exercise the bigger scope of autonomy it has.

Customization

The first level of autonomy that was identified in relation to micro:bit exercises, was allowing pupils to make smaller customization to a predefined design. In the case of a simple "hello world" program, this could mean allowing the pupil

to customize the text string to something else than "hello world". Hence a customization is not something that alters the behavior of a design, but rather allows the pupil to locally modify specific point of interest that have been selected by the person designing the exercise. From Figure 6 it is possible to see that a majority of the choices that have to be made regarding the exercise still has to be made by the teachers when an exercise has this scope of autonomy.
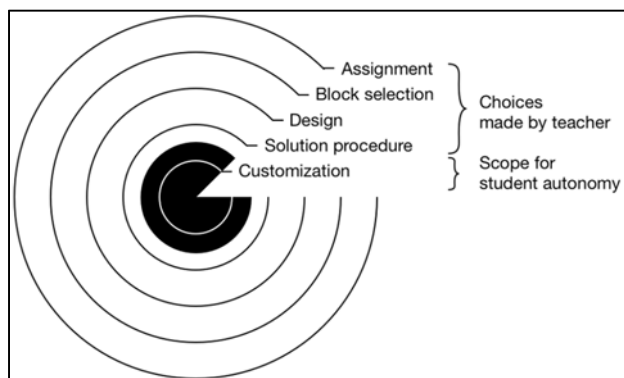
Figure 6: Scope set at Customization level

Solution Procedure

The next level of autonomy that was identified is related to the solution procedure of an exercise. This level of autonomy relates to what subparts of a solution to tackle in what order. When this level of autonomy lies within the pupil's scope, the pupil is free to choose the order in which to create the solution. When the solution procedure does not lie within the scope of the pupils autonomy, the pupils are asked to follow a stepwise procedure instructed by the teacher. In the case of

creating an animation, this could be the difference in starting with drawing the desired animation and then figuring out the best timing between frames, or doing it the other way around. This way there is more freedom for the pupil to make choice about the way they solve an exercise but the target design is still chosen by the teacher, as illustrated by Figure 7.
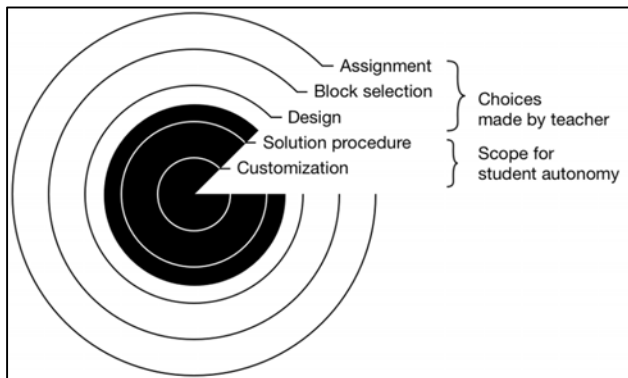

Figure 7: Scope set at Solution procedure level

Design
The third level that was identified is concerned with the design that a pupil makes to complete an exercise. This is a rather interesting level of autonomy, as setting the pupils scope of autonomy to encompass this level means that the teacher no longer knows what the final design will look like, as it is up to the pupil. In contrast to a scope of autonomy that only encompasses the level of customization, a scope that encompasses the level of design allows for completely new design solutions to an exercise, and not only the modification of predetermined placeholders. The pupil is however still restricted by the teachers choice of blocks to be used with this scope of autonomy, as illustrated by Figure 8.
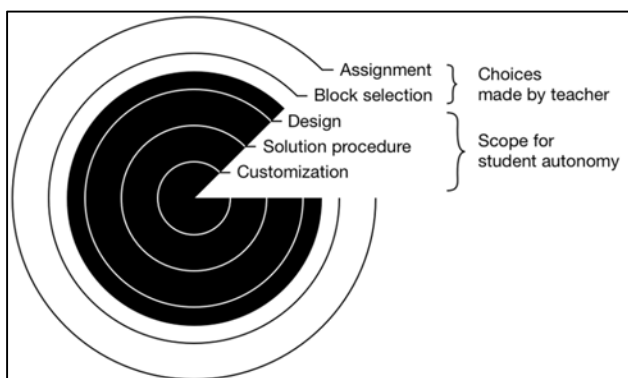

Figure 8: Scope set at Design level

Block Selection
The fourth level of autonomy that was identified is related to block selection, see Figure 9. Blocks are the building pieces that are used to create a design. When this level is not encompassed by the pupils scope of autonomy in an exercise, it means that the teacher has predetermined what blocks the pupil should use to create his or her design.
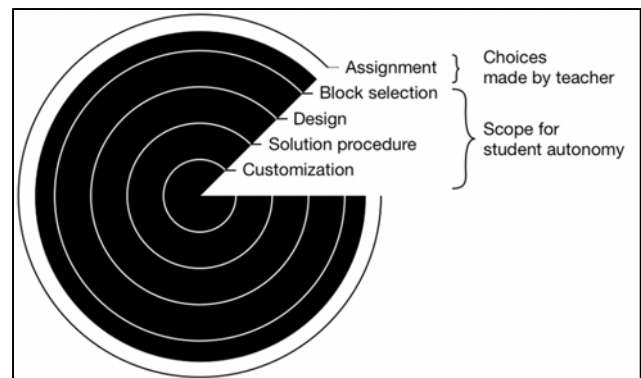

Figure 9: Scope set at Block selection level

This level of autonomy has two parts. The first of which relates to the type of blocks and the second one relates to the quantity of blocks. For instance the teacher can give an exercise where the pupils are asked to create an animation on the micro:bit using any number of blocks of the "loops" and "show LED" variety. This way the types of blocks are chosen by the teacher but the pupil is free to choose the number of blocks. Another exercise could be to make a step counter using only four blocks in total. Here the teacher decides the number of blocks but their type are free to be chosen by the pupil. These two examples illustrate that an exercise can be created in ways where the pupils scope of autonomy only encompasses one of these two block selection levels. Likewise none of them can be encompassed, which means that the teacher decides exactly what blocks ought be used. And lastly when both of them are encompassed by the pupil's scope of autonomy, it means that the pupil is free to create a design out of any block type or quantity, as long as it satisfies the assignment.

Assignment
Lastly, an autonomy level was identified relating to the very assignment itself. This relates to decisions about the topic and aims of an exercise. In the case where this level is not encompassed by the pupils scope of autonomy, the teacher defines what the pupil ought to do in order to complete the exercise. When this level is encompassed by the pupils scope of autonomy however, pupils make the decisions about what the exercise is going to be about. These kind of exercises might initially only be associated with higher educational projects, it is however just as true for exercises where the teacher tells pupils to create whatever they want. Having to create your own assignment is basically the same as having to come up with an original project idea. To be able to handle this level of autonomy, pupils are recommended to have reached a rather high level of experience and be comfortable with making various decisions, or they might run the risk of feeling overwhelmed. As illustrated by Figure 10 this level of autonomy does not require the teacher to make any decisions.
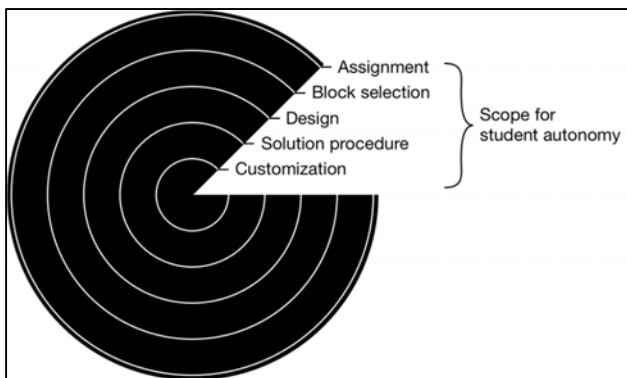
Figure 10: Scope set at Assignment level

DISCUSSION

When creating an exercise for working with micro:bit there seems to be some importance in making a conscious decision regarding its scope of autonomy. The exercise should match the pupil's current level as good as possible, and provide them with an opportunity to improve their independent problem solving skills, without being too overwhelmed. Five levels of autonomy were identified for working with the micro:bit. This set might very well need to be changed or be expanded with more levels. It can for instance be discussed if there are more layers outside of the one called assignment. As a completely autonomous assignment with the micro:bit still is an exercise limited to the hardware micro:bit, it is reasonable to say that there could be a level of hardware and maybe editor outside of the existing levels.

Our experience is that it is beneficial to always provide some scope of autonomy in exercises, as it was seen to pacify pupils when they did not have any way to affect the outcome of the exercises they were doing. In the scope of working with micro:bit, always providing some level of autonomy would translate into always allowing pupils to perform some level of customization in any exercise they are involved in. This relates to Paperts findings that pupils exposed to environments with creative freedom have a higher tendency to learn the necessary knowledge in order to realize their ideas [5].

Pupils are on different levels and require different scopes of autonomy in their exercises, which is a challenge when working with a whole class. As it is hard to give every pupil individually adapted exercises with scopes of autonomy that matches their individual needs, teachers have to find exercises that can be given to the entire class. This means that a class with a wide span in individual progress, a single exercise can be perceived as anything from boring to useful to overwhelming. This can be tackled in various ways. One way is to try to minimize the skill span in the class, and unify the individual levels. A second approach is to expand a single exercise's scope of autonomy to be more flexible. In this way, one single exercise can be given to an entire class, but different modifications or tips can be used to increase or decrease the scope of autonomy for the exercises, to better adapt it to individual pupils.

In Self Determination Theory [8] autonomy is mentioned both as a basic psychological need as well as the causality orientation described as "acting out of interest". This shows that the word autonomy can be a bit arbitrary, and hence might differ slightly from the way it is used throughout this paper. The way it is used in this paper is more relating to the amount of choices available to a pupil in a certain exercise situation. This positions our use of autonomy more as a term relating to the way a pupil's' context, the exercises, can be designed to satisfactorily support both the basic psychological needs of autonomy as well as competence. As it both relates to providing the pupil with enough choices to feel autonomous, yet not provide them with too many choices. As this runs the risk of having them feel that they lack the competence to succeed, which is how the basic need of competence is defined according to SDT [8].

Workshops

By experimenting with different ways of teaching, our experience from conducting the workshops was that co-coding was a useful method as it allows the teacher to probe pupils current skill level with questions and rate the discussions in the class and adapt the level of guidance. Co-coding can be an effective method to teach new concepts, like the need for variables for instance, within the context of an exercise, rather than as a separate presentation. Co-coding should be practiced often as it provides an including activity for the whole class, prepares pupils by carefully giving them new tools to work with and also lets the teacher get an overview of the general understanding towards the subject in the class.

The workshops covered the basic concepts of programming: algorithms, loops, randomness, logic, variables, and debugging. The concepts were introduced and practiced in parallel to each other, rather than separately, in series. This due to the difficulty to create interesting exercises based on single programming concepts, why combinations of multiple programming concepts were practiced.

Generalization

Throughout the workshops, we realize that a majority of the participants had a positive bias towards curriculum changes and digitalization. Therefore, there is no claim that the teachers represent an accurate image of the average mindset and motivation a teacher might have regarding programming.

Throughout the project, we have always related to the basic concepts of programming in our design process. These concepts permeate through all programming teaching activity and promote computational thinking and problem solving. Even though our model was made for micro:bit specifically there are many similarities that makes it versatile. Most of the platforms used to teach programming for primary school uses a block type editor, just like

micro:bit. Other platforms can be better suited to teach some of the concepts in a vacuum, which is not the case for micro:bit, but the workflow still applies as it promotes a progress that aims to keep pupils at a balanced level of stimulation when learning. Since the underlying theories on the behavior of pupils and how they learn is supported in the model, we believe the model can be claimed to have a wider applicability even though quantitative data to support the long term effects of using such model is non-existent in this paper.

Future Work

During our early workshops we encountered the social phenomena of trends among the participating pupils. It began when a pair of pupils programmed a funny game and started playing, others took notice and wanted to join in. Soon the whole workshop had turned focus towards this game and everybody wanted to make it. During this period the pupils were highly involved, motivated and interacted with each other to a great degree, even helped one another. This behavior has been seen multiple times throughout the project, and though it is positive, it seems hard to predict. This could be a topic to investigate further, we believe there is potential to capitalize on if the teacher can seize these moments when trends erupt and adapt the teaching to these opportunities.

The scope of autonomy model suggested in this paper is merely an attempt to explain behaviors and phenomena that were observed throughout the project. A next step would be to see if it is possible to design a study that can validate or falsify the suggested correlation between the amount of choices presented in an exercise, and show the pupils potential to develop problem solving skills and feelings of being overwhelmed. The effect of such model need to be measured in a live setting where conclusions for large sample sizes can be made in order to truly gain any significance.

CONCLUSION

Through an iterative process, a total of 21 workshop interventions were conducted to collect qualitative data and gain insights about Swedish primary school pupils and their teachers interactions with micro:bit teaching materials. The results from the project suggests that it is important for teachers to consider the amount of free choices that are given to pupils in any given exercise, when designing teaching materials with the micro:bit for training primary school pupils computational thinking skills. The contribution of this paper is the Scope of autonomy model, which is based on observed relationships between pupils learning potential, their risk of feeling overwhelmed and the amount of choices provided in exercises. More work is needed to validate the model. The Scope of autonomy model is a result from investigating what is important to consider when designing teaching materials with the micro:bit for training Swedish primary school pupils computational thinking skills, and can be a support for teachers to meet the programming requirements of the new curriculum changes.

REFERENCES

1. Paulo Blikstein. Digital fabrication and 'making'in education: The democratization of invention. *FabLabs: Of machines, makers and inventors*, pages 1–21, 2013

2. Kate Williams. Literacy and computer literacy: Analyzing the nrc's "being fluent with information technology". *Journal of Literacy and Technology*, 3(1):1–20, 2003.

3. Rachel Charlotte Smith, Ole Sejer Iversen, and Rune Veerasawmy. Impediments to digital fabrication in education: A study of teachers' role in digital fabrication. *International Journal of Digital Literacy and Digital Competence (IJDLDC)*, 7(1):33–49, 2016.

4. Regeringskansliet. Stärkt digital kompetens i läroplaner och kursplaner - regeringen.se. http://www.regeringen.se/pressmeddelanden/2017/03/starkt-digital-kompetens-i-laroplaner-och-kursplaner/, May 2017. [Online; accessed 2-March-2018].

5. Seymour Papert and Idit Harel. Situating constructionism. *Constructionism*, 36:1–11, 1991.

6. John AC Hattie and Gregory M Donoghue. Learning strategies: a synthesis and conceptual model. *npj Science of Learning*, 1:16013, 2016.

7. Richard M Ryan and Edward L Deci. Intrinsic and extrinsic motivations: Classic definitions and new directions. *Contemporary educational psychology*, 25(1):54–67, 2000.

8. Edward L Deci and Richard M Ryan. Self-determination theory: A macrotheory of human motivation, development, and health. *Canadian psychology/Psychologie canadienne*, 49(3):182, 2008.

9. Jeannette M Wing. Computational thinking. In *VL/HCC*, page 3, 2011.

10. Shuchi Grover and Roy Pea. Computational thinking in k–12 a review of the state of the field. *Educational Researcher*, 42(1):38–43, 2013.

11. Harvard Edu. Computational thinking with scratch. http://scratched.gse.harvard.edu/ct/defining.html/, 2016. [Online; accessed 2-March-2018].

12. About the BBC Blog Head of BBC Learning, Sinead Rocks. Bbc micro:bit, groundbreaking initiative to inspire digital creativity and develop a new generation of tech pioneers. http://www.bbc.co.uk/mediacentre/mediapacks/microbit/, 2016. [Online; accessed 2-March-2018].