Fostering of interdisciplinary learning through basic education in computer science in mathematics in primary education

Peter Ludes, Marcus Schütte

Fostering of interdisciplinary learning through basic education in computer science in mathematics in primary education

Peter Ludes and Marcus Schütte

Technical University Dresden, Germany; peter.ludes@tu-dresden.de

Digitalization is a topic that is ubiquitous in everyday life. The technological revolution progresses with high pace. Technology/web/mobile companies all over the world are flourishing but primary schools (in Germany) do not follow the economy's development and has not made computer science part of its curriculum yet. This is either due to technical equipment limitations or to the inability or insecurity of teachers to include technology and the underlying principles in their classroom. Furthermore, it is not officially integrated into the core curriculum which prohibits the implementation of a universal standard for computer science competences that have to be taught in primary schools. The paper will present the underlying project and the development of specific learning environments that are designed to nurture cooperation and interaction between participants to have a closer look at the in-between, where learning occurs. This will help to solidify the claim that computer science in primary school has a right to exist.

Keywords: Computer science, interaction, mathematical competences, digitalization.

Introduction

Computer Science, digitalization or technology are only three out of many other terms concerning the digital age that ubiquitous in today's society. Nearly every professional branch has to deal with technology and digitalization at some point. Employees have to learn how to deal with these digital structures mainly on their own, because they never experienced a fundamental education in computer science. Many schools in Germany offer Computer Science only as a subject in secondary school although coding is seen as the language of the 21st century. Whoever is able to "speak" a programming language will be understood all over the world. Many IT unicorn businesses have their roots in a teenager's room or the parents' garage. Often, these businesses came to be because exceptional skill and auto didactical training come together. The masses only seem to use the technology that emerges from these businesses without understanding the basic underlying structures that enable the creation of a product. No matter which idea came to be, it has almost certainly been realized through programming. The choice of programming language is rather secondary as they all follow the same basic rules. Children, from first grade onwards, know about digital technology and computers in different shapes and formats, but when they enter school, these digital tools often cease to exist. Therefore, a responsible use of these technologies and learning in computer science is not fostered in primary schools. Tablets, laptops, smartphones and gaming consoles are at least present once in every household. According to the KIM study, most families own more than one of these devices and use them more than once a week (e.g. the percentage of smartphone/laptop ownership is 97%). In fact, 41% of children from 6-10 own a smartphone themselves. Children learn from an early age how to use e.g. tablets. The functions of these devices are often self-explanatory. The underlying structures, e.g. how is software developed, what rules does a computer follow or how to troubleshoot problems, are not that clear to most children although these structures are relatively similar across all digital devices, this also is true for most teachers, who did not grow up with technology and are now supposed use it in the classroom and teach its responsible use. Compared to mathematics this would be similar to learning specific calculations with specific numbers or materials but not understanding how to transfer and apply e.g. long division to other tasks. The paper does not aim to defy media competences and how they can be accomplished, it is rather a claim to go further and to embrace computer science itself into primary education and see where and when interaction and negotiation of meaning between children occurs and how it fosters the learning of computer science and mathematics competences. This would empower children to accomplish much more with their knowledge about media and their use (knowledge, they undoubtedly gained while growing up in a digital world).

Status Quo: Media competences, a watered-down term

Computer Science is a topic that is present in almost every secondary school, either as a mandatory or a facultative subject. Either way, the subject is taught by teachers, who have a higher education degree in computer science. To become a primary school teacher in Germany, as in many other countries, the selection of specific subjects is not necessary, as the primary teacher is trained to teach all subjects that are part of the primary school curriculum (although many universities offer the opportunity to choose a core subject (e.g. mathematics) that will give the student the ability to enjoy an even more focused education in this specific area. Computer Science as a subject is not present in either primary school or primary school teacher education. Future teachers rather can archieve something that is often called media certificate or media training. Mostly, this is integrated into the courses that are already taken at university. E.g. one course would include a topic like "use of digital media in geometry in primary school". This is neither standardised, nor does it aim at specific competences. It is rather a subject specific realization of the use of technology for a certain topic. Today, most schools decide themselves how and when to use digital media and often one feels that it is mostly aimed at helping teachers to keep up with what children already know. If they decide to do so it is often done under the term of media competency. Krauthausen (2012) mentions that the term "media competences" is colorful and can be interpreted very differently according to specific interests. For some schools media competency means that the teacher uses e.g. an iPad, for others that the children learn how to use word processing software. Other projects try to foster the use of specific types of media such as podcasts to facilitate learning process for mathematics (cf. Schreiber, 2012). This is far away from being similar to what computer science would request children to learn. Computer science is considerably more than just knowing about the functions of a computer. It is learning about logic, about algorithms, about programming & robotics and (late-breaking) cryptography. Questions like: *What is an algorithm? What is logic? What technology can I use to facilitate my work?* can be answered by young children to improve their learning. But not only will the acquisition of such skills foster the expert knowledge, it will also nurture competences that can be interdisciplinary used. This is important, as we have to ask whether our traditional cultural competences are still sufficient in a more and more digitalized world or whether we need to teach computer science as well. Some projects try to accomplish this, e.g. Herper and Hinz (2009) through computer science education in primary school and Weigand (2009) with his project: "Algorithms in primary school" (title translated by the author). Weigand especially shows that to work on basic principles of computer science (in this case algorithms) a computer itself does not necessarily have to be available, as he uses pen and paper to work on algorithmic processes. Primary schools do not offer the subject of computer science and it will not be easy to implement it into the curriculum as an independent subject. To achieve this goal, a back door has to be found to sneak computer science into primary school. Then, once its benefits have become obvious the step from being part of another subject to being an independent subject is only a small one. The answer to the question what subject should be used to include computer science in its contents could not be answered more easily. Mathematics seems to be the ideal candidate, as competences in both subjects are similar right up to identical. We will now focus on these similarities in more detail.

Competences in Mathematics and Computer Science

The German core curriculum provides two types of competences for the subject of mathematics: the general mathematical competences and the content-related mathematical competences (KMK Bildungsstandards, 2005). General mathematical competences include arguing, problem solving, communicating, modelling and the presentation of mathematics. Although these competences will equally be taken into account, the main focus here will be placed onto the content-related mathematical competences, which are numbers and operations, space and shape, pattern and structures, sizes and measurement and data, frequency and probability. These content-related mathematical competences provide the framework for the contents of the learning environments. To justify these contents, a side by side comparisons of some chosen content-related mathematical competences and where to find them in computer science will be done. The mathematical competences will be presented as written down in the German core curriculum, then computer science content that also matches these competences will be provided.

Mathematics	**Computer Science**
Understand the relationship between and the representation of numbers, Understand and master calculations	Algorithms use calculations, Loops have to be counted, Sorting algorithms, Types of variables integer, float, double

Table 1: Competences in Mathematics and CS: Numbers and Operations

Mathematics	**Computer Science**
Spatial orientation	Program robots, Define an area of movements,Plan with obstacles and predict motion sequences

Table 2: Competences in Mathematics and CS: Space and Shape

Mathematics	**Computer Science**
Recognize and characterize regularities	Structure and plan algorithms, Plan processes and translate them into a programming language, logic, Sorting algorithms

Table 3: Competences in Mathematics and CS: Pattern and structures

Mathematics	Computer Science
Have the ability to imagine sizes, Have the ability to use sizes in specific situations	Determine the step range of a robot, Determine run time

Table 4: Competences in Mathematics and CS: Size and Measurement

Mathematics	Computer Science
Understand the relationship between and the representation of numbers, Understand and master calculations	Algorithms use calculations, Loops have to be counted, Sorting algorithms, Types of variables integer, float, double

Table 5: Competences in Mathematics and CS: Data, Frequency and Probability

(KMK Bildungsstandards, 2005)

This selection of competences from the core curriculum shows that to each content related mathematical competence a related topic in computer science can easily be found. This is highly interesting, as it suggests that mathematical competences are similar to those that are required to perform tasks in computer science. Thus, mathematics can fulfil the requirements to integrate elements of computer science into its curriculum. To prove this claim, it would hardly be possible to modify the existing mathematical lessons to include this content. Rather, specific learning environments have been developed to show that learning computer science topics nurtures the competences that are necessary for both mathematics and computer science.

The pilot project

Partner school

The search for partner schools was far from easy. Many schools were not interested. One school on the other hand was immediately willing to participate and upon further information managed to interest 19 children from grade four to take part in the project. As we did not expect such an overwhelmingly large number of participants from one single school, we cancelled all further efforts to acquire more schools and decided to work exclusively with the just one partner school. The school was very open to our project and supported us from day one. The first feedback suggests that all children are highly enthusiastic and motivated. A claim that we can only support after our first three weeks. In agreement with the school, we chose two time slots of 90 minutes in the afternoon twice a week (Tuesday and Thursday) over the course of six weeks. During the time slots the children worked on the learning environments in pairs (some tasks required two groups to come together to discuss) supervised by a student from the seminar or Peter Ludes.

Learning environments

To examine whether certain competences are used during working with tasks, specific learning environments have been developed that have a computer science topic as a core topic. The learning environments have been developed during one of Peter Ludes' empirical seminars for future elementary school teachers. The students developed the first draft of the learning environments on their own and after a review process finalized them collectively during the seminar. The main topics are: logic (general and propositional), algorithms, cryptography and programming/robotics. The main challenge for the students was to develop learning environments with core topics that are not (yet) part of their actual studies, as computer science is not a topic that is taught in elementary school. Therefore, an extensive introduction into the field of computer science has been necessary.

A second demand that had to be kept in mind during the development has been a focus on interaction. The tasks should be designed in a way that -at least partially- fosters interactional processes between the participants as we focus on learning through collective argumentation and participation (Krummheuer, 2011). The first learning environment focused on the topic of logic. The primary task always aims to get a first impression of what the participants already know (or think to know) about the concept. This very open question (e.g. *What does logic mean?* or *When does a person have to think logically?)* provides a wide variety of possible answers without any pressure for right or wrong. The supervising students are always advised to let the children speak freely as much as possible, unless a lively discussion does not occur, general questions or guidance is to be avoided. Although the content is important, the actual focus for this pilot project is not to survey content learning, but rather the learning that occurs between individuals whilst discussing and arguing about the specific topic. Learning in primary school is initially dialogical learning. That is, learning is seen as a dialogical process in contrast to learning as a monological process, which would be rooted in the individual (cf. Miller, 1986). The learning of mathematics can be seen as an increasing autonomous participation in collective argumentations that are produced and nurtured collectively by the group itself (Krummheuer, 1992; Voigt, 1995). This idea of learning can be transferred to the learning of basic competences in computer science. To build upon this concept, all ways of communication and interaction between the children has to be supported as much as possible. To ensure an efficient way to videotape the children working on the learning environments, we designed the environments to be worked on in pairs or groups of three, sometimes with a closing task, that included a larger group discussion in groups of up to six children. Working in smaller groups for us provides the advantage, that single children are not able to extract themselves from group tasks or discussions but rather encourage them to participate. Every learning environment is designed to cover three timeslots of 90 minutes. Here, two sessions of 90 minutes are planned for the actual content-related activities and one session of 90 minutes reserved for documentation and evaluation of learning processes though tools like learning diaries, wikis and storyboards. Learning diaries can e.g. be used to recapitulate the learning process, correct misconceptions and enables the child to visualize what learning progress it has made. We chose the storyboard as an adequate tool for the programming/robotics learning environment. This environment will be realized with LEGO Mindstorms EV3 Educations sets. The main goal after developing and building the robot itself will be to program its specific actions. These actions should be planned beforehand because the children can choose from a variety of actions and sensors with endless combinations. For this task, a storyboard is the ideal candidate as it enables the children to structure their thoughts and plan the movements and sequences that the robot has to fulfil. It also makes trouble-shooting rather easy as the children can always compare their plan to the robot's actual movements and actions.

Analysis

The analysis of the results of the work on the learning environments is based on methods of interpretive classroom research, such as interaction analysis. As we place much importance onto interaction, cooperation and discussions this method seems most appropriate to us. It helps to find more suitable tasks that help children to collaborate in a productive way. The negotiation of meaning is a key element that has to be focused (cf. Krummheuer, 1992). How does the negotiation of meaning in collective processes of argumentation occur and how can it be supported thought the task itself? According to our perception, learning does not occur inside the individual but during the interaction between individuals, whilst discussing, talking, arguing and also justifying the own answers and the answers of others (cf. Krummheuer, 1992). The underlying concept is founded on the ideas of the symbolic interactionism (cf. Blumer, 1969) that will allow us to examine learning as the increasing autonomous participation in collective argumentation in computer science discourse. Possibly this could lead to the definition of participation profiles (cf. Brandt, 2004) for different students, specifically tailored to the computer science classroom. This could not only benefit primary education, but also computer science classrooms in secondary schools. If a specialized and standardized computer science education in primary school was mandatory, secondary schools would have a contact point and a profound basis on which they could build and focus their curriculum.

First impressions

The first impressions of the pilot project are consistently positive. One major question during the development process was whether children could be unchallenged or overwhelmed with the tasks as they work with topics that are not being taught in primary school and are therefore unfamiliar. The individual knowledge of the children concerning this specific topic could therefore be developed very differently. The children engaged immediately with the tasks and lively discussions occurred. In the very beginning, a certain insecurity was noticeable. This was expected as the children naturally try to give correct answers. This was not possible as most questions are designed in a way that multiple answers and also very individual answers are possible. After the first sessions, the children got more and more used to the types of tasks and felt more comfortable when joining the discussions.

Interestingly, mathematical discussions occurred very often, whether they were planned or not. One example where a rather simple question led to a lively discussion occurred during the learning environment: logic. The children had to decide whether a statement is right or wrong. One of the statements claimed: If something is round it is not pointy. Right or wrong? The children then engage into a discussion, where they have to decide what qualifies as being round:

S1:	If something is round it is not pointy\ [reads]
	That does not have to be true\
S2:	(It is not\) [laughs].
S1:	That does not have to be true maybe it is so to say a pointy circle\
I:	How does a pointy circle work?\
S1:	So . wait (4) [draws into his folder]
S2:	[looks into S1’s folder]

S1: Like this and so on and on \ [points at the drawing in his folder]

S2 Yes, but if something is round [draws a circle into the air with his pen] so this here [points at S1's drawing] that is not round\ that is pointy\

S1: Yes, it is **roundly** but fi it is round then you are right\

First, S1 states that he is not convinced that this statement is true and he tries to find a shape that is pointy and round. He therefore draws a shape that has corners but is not just one straight line, similar to a part of a polygon. S2 then looks at the drawing and defines the shape as pointy because it has corners. S1 then again discusses the word round and seeks for a better word to describe his drawing and proposes the word roundly for something that is not a straight line but follows the form of a circle part although it has corners. S1 and S2 clearly have a dissent in what qualifies as being round and then shift this disagreement into a consensus through the introduction of a new vocabulary. It is rather interesting that the S1 and S2 discuss to find a consensus and the questions remains, what would happen if they could not find one. Here, the material itself proposes a research question: How do consensus and dissent influence cooperative learning. Is learning through collective argumentations also possible on the basis of a dissent or does a consensus have to be reached in order to complete/move on with the task itself? The structures in cooperative learning opportunities could be fundamentally different if the necessity of reaching an agreement would not be mandatory. This question will be focused during further analysis as the concept of computer science learning environments provides a perfect frame: It has clear and visible connections to mathematics but is fundamentally new and different, so that children can learn a new topic in which they do not have prior knowledge to build on.

Prospects

The learning environments with the topics cryptography, algorithms, programming/robotics are also completed by now. The videos and writings of the children are being extensively analysed using methods of interpretive classroom analysis. The analysis will enable us to look at the negotiation processes in more detail and determine, where the learning of individual computer science concepts and understandings occurs and how meaning is negotiated during the task itself also in regard to cooperative learning and the underlying structures. Through this, it will be possible to rework the learning environments to tailor them even more specifically to their purpose. The reworked learning environments will then be used in a more extensive main study to examine our claims and to strengthen the position of computer science as a key part of a profound and forward-thinking education that will not only benefit children's abilities in this specific subject but also strengthen their mathematical competences

References

AKBSI - Arbeitskreis „Bildungsstandards" der Gesellschaft für Informatik (Eds.). (2008). *Grundsätze und Standards für die Informatik in der Schule—Bildungsstandards Informatik für die Sekundarstufe* 1. *LOG IN*, (*28*)150/151, Beilage.

Blumer, H. (1969). *Symbolic interactionism*. Englewoods Cliffs, NH: Prentice-Hall.

Brandt, B. (2004). *Kinder als Lernende – Partizipationsspielräume und - profile im Klassenzimme: Studie zur Partizipation im Klassenzimmer* Frankfurt: Lang.

GI – Gesellschaft für Informatik (Eds.) (2006). *Was ist Informatik?* [Unser Positionspapier]. Bonn.

Herper, H. & Hinz, V. (2009). Informatische Bildung im Primarbereich. In Koerber, B. (Ed.), *Zukunft braucht Herkunft* (pp. 74-85). Bonn: Gesellschaft für Informatik..

Medienpädagogischer Forschungsverbund Südwest (2015). *KIM-Studie 2014: Kinder und Medien; Computer und Internet.* Retrieved from http://www.mpfs.de/fileadmin/KIM-pdf14/KIM14.pdf

KMK - Kultusminiserkonferenz (Ed.). (2009). Empfehlung der Kultusministerkonferenz zur Stärkung der mathematisch-naturwissenschaftlich-technischen Bildung. Retrieved from http://www.kmk.org/fileadmin/Dateien/veroeffentlichungen_beschluesse/2009/2009_05_07-Empf-MINT.pdf

Krauthausen, G. (2012). *Digitale Medien im Mathematikunterricht der Grundschule*. Berlin & Heidelberg: Springer-Verlag.

Krummheuer, G. (1992). *Lernen mit" Format": Elemente einer interaktionistischen Lerntheorie; diskutiert an Beispielen mathematischen Unterrichts*. Weinheim: Deutscher Studien Verlag.

Krummheuer, G. (2011). Representation of the notion “learning-as-participation” in everyday situations of mathematics classes. *ZDM*, 43(1/2), 81−90.

Miller, M. (1986). *Kollektive Lernprozesse*. Frankfurt: Suhrkamp.

Schreiber, C. (2012). Podcasts selbst erstellen? Na klar: PriMaPodcasts! *Grundschulunterricht Mathematik*, *4*, 39−42.

Ständige Konferenz der Kultusminister der Länder in der Bundesrepublik Deutschland (Eds.). (2005). *Bildungsstandards im Fach Mathematik für den Primarbereich (Jahrgangsstufe 4).* München, Neuwied: Wolters Kluwer Deutschland GmbH.

Voigt, J. (1995). Thematic patterns of interaction and sociomathematical norms. In P. Cobb & H. Bauersfeld (Eds.), *The emergence of mathematical meaning: Interaction in classroom cultures* (pp. 163 – 201). Hillsdale, N. J.: Lawrence Erlbaum.

Weigand, M. (2009). Algorithmik in der Grundschule. In B. Koerber, B. (Ed.), *Zukunft braucht Herkunft* (pp. 97−108). Bonn: Gesellschaft für Informatik.