PHYSICAL COMPUTING SYSTEMS – A SYSTEMATIC APPROACH

Eric Schätz and Alke Martens[1]
University of Rostock, Germany
[1]*Prof. Dr.*

ABSTRACT

As we talked with school teachers about the use of physical computing systems in class, one major drawback of these systems became obvious: almost nobody – next to enthusiastic autodidacts – has been able to tell us on an abstract basis for which educational purposes a certain physical computing system can be used, i.e. in a goal-oriented way. This insight lead to the approach to structure the field of physical computing systems with the overall goal to make it easier for teachers to improve their lectures due integrating physical computing systems into computer science education. The aim of this paper is to structure the field, to find categories and to show how existing devices can be integrated into the founded structure.

KEYWORDS

Physical Computing, Computer Science Education, Categories

1. INTRODUCTION

In the field of computer science education, teachers are confronted with a quite diverse situation. On the one hand, they find pupils which are still completely new to computer science and think themselves naive in this context. On the other hand, they find kids, which call themselves nerds and think that they are hackers yet. Thus, a classroom-oriented education, where this plethora of individuals can be found, can be a really demanding task. Being researchers in the field of computer science education at school (in Germany this is called Informatics Didactics, in some language regions better sketched as Instructional Design for Informatics), we have observed different reactions to these difficult situations: some teachers reduce computer science education to programming, some teachers really focus on the theoretical aspects of computer science and completely neglect the practical parts in computer science education, just to sketch some extremes. Coming from the mindset, that *constructivist learning* is an important approach about learning, we have tried to offer workshops about physical computing systems, as we have learned in working with school kids that almost all kids from the scale of naive users up to ones who are really interested in computer science have fun and a really good and continuous high level of motivation while working with these systems. Being aware of the plethora of physical computing systems available on the market, we will make clear that we are only focusing on physical computing system which can be used for educational purposes (cf. (Hodges, Sentance, Finney & Ball 2020)).

In contrast to our observation with school kids, we have to admit, that there is a gap between the oral feedback we get from teachers right after a workshop and the one we get some weeks later about the teachers' intentions to use physical computing systems in their lectures. This is also reflected by other researchers in the field, e.g. (Schmalfeldt, Thomas & Przybylla, Mareen 2021), and (Mareen Przybylla, Ralf Romeike 2017). Thus, we interviewed several teachers to find out what might be the hindrance of integrating physical computing systems as one part of their educational spiral. In the result, we got the insight, that there is a certain barrier in using these tools: there are so many of them, it is not clear how they can be used in a goal-oriented way, they are inherently complex and not that easy to handle from the computer science perspective. Plus, the really "freaky" ones cost a lot of money. Even if the last point could be handled by using cheaper and more flexible devices, like the BBC Micro:Bit (Microbit Foundation 2020), the problem there is that teachers feel that there is a lot of work required before they are able to use them in class.

Thus, our approach is to firstly investigate existing systems, secondly to develop a way to organize them, i.e. categories for systems, which in the best cases can also be used for future systems, and thirdly, to connect the pedagogical planning process (which is the process of planning a lesson) with the developed categories.

In this paper we are going to focus on the first two steps. In the following we are going to show how we have structured the field, which aspects we have found most important for our target group and how we have realized them. The remainder of the paper is organized as follows: in the next section, we will show some background of concepts of physical computing systems, then we will demonstrate the categories and the attribute system and sketch its relation to the pedagogical planning process. We will give some examples, how devices fit into the attribute system. The paper closes with a discussion and an outlook.

2. STATE OF THE ART

Physical Computing Systems (PhCS) are a field which developed really fast over the last years – comparably all the new technologies available have been developed in the last ten years, even if the field can look back on a comparably long tradition (O'Sullivan & Igoe 2004). Thus, it can by all rights be called a comparably young field of research. Only few approaches exist, which help to structure the field (e.g. (Mareen Przybylla, Ralf Romeike 2017), and (Przybylla & Grillenberger 2021).

But which gap are physical computing systems closing, when they are embedded in the educational processes at schools and why should this be done?

This question is answerable on two levels: first of all, PhCS are surprising, non-conventional, not obviously found everywhere and they are no smartphones or the strange boxlike things the adults work with every day. Most of them are way less complex as high-level computer systems like laptops or smartphones are and they contain less black-boxes, a term we are going to discuss in an upcoming paper. Thus, school kids tend to be surprised when working with PhCS in educational school setting, as they in most cases have not done this before and somehow these things are what kids call "interesting", as they come with the feeling of a playful approach. PhCS can come in the shape of a controller board like the BBC Micro:Bit (Halfacree 2017) or a Calliope Mini (Calliope gGmbH 2022), or in the shape of a small robot, like for example the Anki Cozmo (Wilhelm 2021). On the other level, these systems are in most cases no complex computer systems, like a "real" computer or a laptop, but more like "computing systems" (Assaf 2018), which means that you have to do more to get the system running, but you find less overload, less unimportant things, and in the end kids have the chance to put their fingers directly onto the point where programming, algorithmic thinking have to take place, where input and output are clearly connected and are not processed somewhere mystical (O'Sullivan & Igoe 2004) (Hüwe & Hüwe 2019). Additionally, for the named examples but also for almost all the systems we have investigated, it can be said that multilevel programming interfaces are available, with a range from very simple and intuitive block like languages (like Scratch, (Scratch Foundation 2022)), more complex ones, and finally "real" programming languages like Python or Java for example.

Naturally, with these systems, teachers have to be really aware of which aspect they want to mediate – i.e. do they want to teach algorithmic thinking and shift to a PhCS after doing this in an unplugged way, or do they want to teach connections of items in a digital world. Related to the goal of the educational setting, different physical computing systems are best suited to this approach. For example, the BBC Micro:Bit comes in the shape of a tiny controller board, which has to be connected to something, e.g. a computer or at least a battery package. When connected to sensors, the program has to be loaded on the BBC Micro:Bit to be available. But how can this take place? So instead of teaching aspects of computer boards on a theoretical basis, like processing unit, storage, input and output, an educational question might be: how can the information be brought to this tiny mini-computer? What is required? Later the kids learn that at the core, all the "big computers" work in a comparable way. Another example might be to actually teach programming: kids, and admittedly also students in their first semesters, as we found out, really loved to get a Cozmo or a Lego Mindstorms robot (LEGO Group 2022) running and doing its tasks, instead of having a console program doing something with code and numbers. This observation is valid for all genders and has been part of gender related research (e.g. (Katterfeldt, Dittert, Ghose, Bernin & Daeglau 2019)).

However, the gap between the assumed advantages of PhCS in educational settings and the amount of systems found at schools and used in educational settings are divergent. Thus, in our project we tried to put the finger on the reasons for it and, surprisingly, it is not (only) the technical barrier and the lack of time (or money) but also that teachers in some cases do not know about the full educational potential of a certain system. Following these insights, gathered by interviewing teachers in federal state Mecklenburg Western Pomerania, Germany, we put the focus on the PhCS, and developed a systematic overview, based on categories and attributes.

3. CATEGORIES AND ATTRIBUTES

In the planning process of a lecture (e.g. in (Meyer 2015), (Heckmann & Padberg 2012) page 114), the teacher goes through these phases:

- content analysis (theoretical/ scientific background of the content of the lesson),
- organizational analysis (how is the situation in the classroom regarding school kids and technical stuff alike),
- educational analysis (which parts of the topic are important, which can be reduced) and finally the
- methodological analysis (how the teaching takes place, how shall the kids work, etc.).

Here, the content analysis is also our starting point, where the first categories and attributes of the PhCS are needed. Thus, with the background knowledge of the pedagogical planning process in mind, we started to look deeper at the PhCS. We investigated a number of 7 PhCS, and came to the following result: We found 11 categories, under which we can focus certain aspects of a PhCS, and related attributes for each of the category. The attributes are sometimes used like a range or a scale where the named attributes denote the upper and lower boundary of the range.

1. Hardware focusing – attributes: usability with periphery, without periphery
2. Form of the representation of supported programming languages – attributes that describe the supported programming languages, for example graphical or textual language
3. Processing degree of the actuating components – range between technical and physical borders of the hardware, range between levels of quality borders
4. Processing degree of sensor components – range of sensor quality, attributes denoting technical and physical borders of hardware components
5. Supported programming paradigm – attributes: imperative, state oriented, logical, object oriented (mixed attributes are usable)
6. Robustness against wrong handling – attributes with related ranges: ranges for connectors and ports, ranges for software robustness, ranges for robustness of the case (of the device) e.g. against dropping
7. Openness of the ecosystem – attributes: to other manufacturers, open source, etc. (there might still be other items to be found, which we have not located yet)
8. Extensibility – range in software extensibility and range in hardware extensibility
9. Administrational aspects – range of inventory administration effort of hardware, range of providing of software modules, range in effort for putting into service
10. Price – attribute absolute price, range, attribute or matrix: price compared to other opportunities
11. Availability of supporting systems – range of documentation from the manufacturer, range for community, attributes availability (and potentially also quality) of lecture material

As readable in the list, our categories start from all the well-known places. We came from the hardware, went over to the software and then focused on other aspects as well. As the system has been developed of an analysis of numerous available PhCS in our lab, we claim that upcoming PhCS can be described with attributes from the developed categories. As an example, we show in the following picture (figure 1) how the category “Robustness” with an exemplary range applied to the complete system can be displayed in a computer-based system helping the teachers to find their best suited PhCS. The PhCS investigated here is the BBC Micro:Bit (cf. (Microbit Foundation 2020)). In the blue box, the overall robustness of the device is shown – the Micro:Bit is one of the most robust PhCS in our collection. However, regarding the range, we can make a more detailed description regarding the overall robustness of the Micro:Bit with periphery and the Micro:Bit without periphery. Pictures of the devices are added to the figure: on the left side the Micro:Bit without periphery is shown, on the right side, there is the Micro:Bit with periphery (just as an example). The range makes visible, that the Micro:Bit itself is very robust (orange scale in figure 1), but the more peripherical add-ons are used, the less robust the system becomes (grey scale in figure 1).

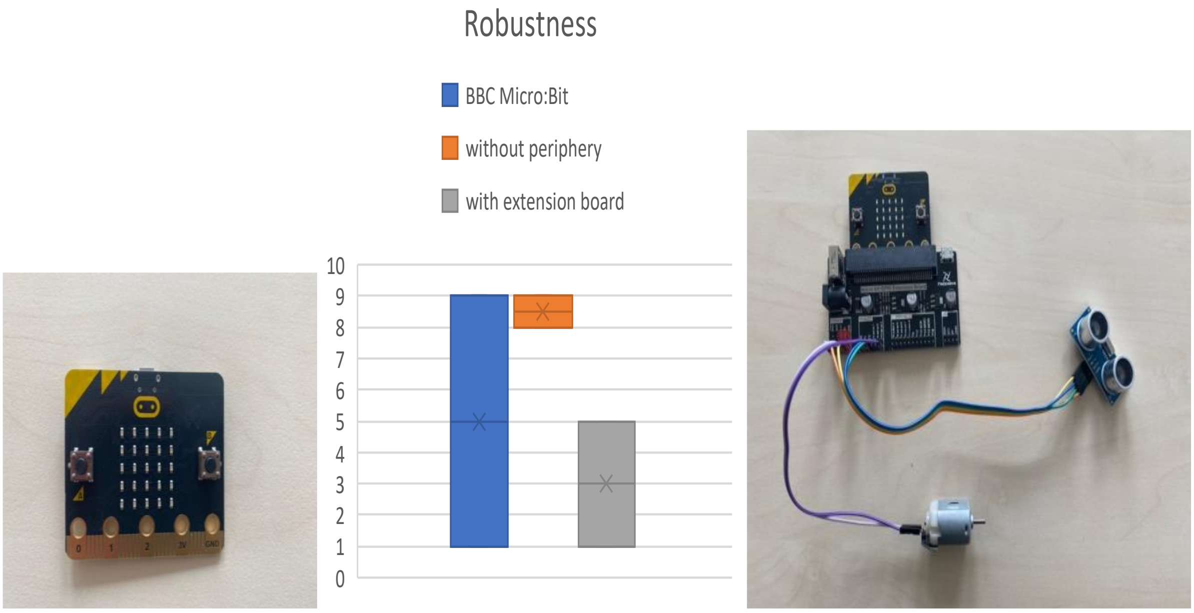

Figure 1. Microbit and the category “Robustness”

As a next step, we made an approach to integrate the categories and their relevance in each of the steps in the above-mentioned planning process. The result is shown in table 1. In this table, the entries +, o, and – show, how important the according category is for the related step in the planning process, meaning + = important, o = semi, and **-** = not important. The table can be read as follows (again with the robustness example and the Micro:Bit): Regarding Robustness, those attributes are not important for the content analysis but really important for the organizational analysis. In case of the BBC Micro:Bit, which is really robust without periphery and can be rather fragile while using an extension board, it should used in a way, the behavior of the students in the analyzed class allow. So if there is a big number of students which tend to ignore instructions, the Micro:Bit can only be used without external periphery.

Table 1. Relation between Attributes and Pedagogical Planning Process (Schätz & Martens 2022)

	Content analysis	Organizational analysis	Educational analysis	Methodical analysis
Hardware focusing	0	+	+	0
Form of the representation of supported programming languages	+	0	+	0
Processing degree of the actuating components	+	-	+	-
Processing degree of sensor components	+	-	+	-
Supported programming paradigms	0	0	+	0
Robustness against wrong handling	-	+	-	+
Openness of the ecosystem	-	+	-	0
Extensibility	-	+	0	0
Administrational aspects	-	+	-	+
Price	0	+	-	-
Availability of supporting systems	+	0	-	0

Based on this table, an overall hint towards usability of PhCS can be derived.

The categories, attributes and the investigated system shall as a next step be transformed to a digital version, which can be used in the pedagogical planning process by the teachers. Also, we are going to investigate more concrete PhCS. As a result, teachers can decide which PhCS have the potential to increase the quality of a specific lesson based on the shown table and the results we get from our investigation (which we are going to provide to teachers as well) during their lecture planning process.

4. DISCUSSION AND OUTLOOK

Computer Science education is not state of the art in all parts of Germany. Due to federalism, the content of the school education system is slightly different in all German federal states. In Mecklenburg-Western Pomerania for example, the government is proud to say that their country has the longest tradition of school education with computer science as a full-fledged topic, starting at the fifth grade (kids are around 11 years old), and with the possibility to have this topic till the end of school time (e.g. baccalaureate) (Schwarz, Richard, Hellmig, Lutz & Friedrich, Steffen 2021). Thus, there is a number of really motivated teachers, which are comparably good supported by their schools. As a result, bringing the idea of physical computing systems to the school falls on a fruitful ground in this federal state. However, we were surprised that even over the years, all the offerings, e.g. workshops and other forms of vocational training, have not led to a raised level of teachers using PhCSs in computer science classes in more situations that really needed.

Based on the combination of the attribute system with the planning process, we hope that we might fill the gap. Admittedly, integrating a PhCS into a school setting is still a demanding task for a teacher and requires a lot of basic knowledge, which the teachers have to take care of in their free time. It is, thus, part of the lesson planning process, but, like all the computer science stuff, PhCSs are susceptible for failure (let it be hard- or software), require a stable internet connection and shall best not be thrown off the table too often. In addition to the integration into the planning process, we advise our computer science teachers to be, that they best always have a plan B in their pockets when it comes to using any device in computer science lectures. This is valid for full-fledged computers, but even more for PhCS.

Regarding the attribute system and the categories, we are optimistic that this will help the teachers to at least think about using the PhCSs in their lectures, as they can see that these systems can be integrated smoothly. First tests have shown promising results, which are as the next step now put into an empirical evaluation, which has been started and will run for the next two years. Looking at our teacher students, the system has proved its viability, yet.

Another quite important aspect of this system of categories might be the idea, that so many new PhCSs have been developed over the last years, not all of which have a clear purpose, but many of which come with the promise, that programming can be learned. The system of categories with the focus on educational settings allow for companies, to check their newly developed system-to-be regarding the fitness for educational settings and hopefully, sooner or later, PhCSs especially for educational purposes will be developed which are really supportive, enjoyable and useful for school settings.

REFERENCES

Assaf, D., 2018, viewed 18 July 2022, from https://mia.phsz.ch/pub/Informatikdidaktik/PhysicalComputing/PhysicalComputing_PHSZ_DA_V2.0.pdf.

Calliope gGmbH, 2022, ‘Calliope’, in *Calliope*, viewed 18 July 2022, from https://calliope.cc/.

Halfacree, G., 2017, *The Official BBC micro:bit® User Guide*, John Wiley & Sons, Inc., Indianapolis, Indiana, doi.org/10.1002/9781119413752.

Heckmann, K. & Padberg, F., 2012, *Unterrichtsentwürfe Mathematik Sekundarstufe I*, Spektrum Akademischer Verlag, Heidelberg, doi.org/10.1007/978-3-8274-2934-6.

Hodges, S., Sentance, S., Finney, J. & Ball, T., 2020, ‘Physical Computing: A Key Element of Modern Computer Science Education’, *Computer*, 53(4), 20–30, doi.org/10.1109/MC.2019.2935058.

Hüwe, P. & Hüwe, S., 2019, *IoT at Home: Smart Gadgets mit Arduino, Raspberry Pi, ESP8266 und Calliope entwickeln*, (#makers do it), Hanser, München.

Katterfeldt, E.-S., Dittert, N., Ghose, S., Bernin, A. & Daeglau, M., 2019, ‘Effects of Physical Computing Workshops on Girls’ Attitudes towards Computer Science’, in *Proceedings of the FabLearn Europe 2019 Conference*, ACM, Oulu Finland, pp. 1–3, doi.org/10.1145/3335055.3335066.

LEGO Group, 2022, ‘MINDSTORMS EV3 | MINT ab der neunten Klasse’, in *LEGO® Education*, viewed 21 May 2022, from https://education.lego.com/de-de/products/lego-mindstorms-education-ev3-set/5003400#ev3-set.

Mareen Przybylla, Ralf Romeike, 2017, ‘Von Eingebetteten Systemen zu Physical Computing’, 10.

Meyer, H., 2015, *Didaktische Modelle: Buch als E-Book*, Verlag an der Ruhr, Berlin, GERMANY, viewed 10 May 2022, from http://ebookcentral.proquest.com/lib/ubrostock-ebooks/detail.action?docID=2080721.

Microbit Foundation, 2020, ‘The new BBC micro:bit with speaker and microphone’, in *New micro:bit annouced*, viewed 21 May 2022, from https://microbit.org/news/2020-10-13/meet-the-new-bbc-microbit/.

O’Sullivan, D. & Igoe, T., 2004, *Physical computing: sensing and controlling the physical world with computers*, Thomson, Boston.

Przybylla, M. & Grillenberger, A., 2021, ‘Fundamentals of Physical Computing: Determining Key Concepts in Embedded Systems and Hardware/Software Co-Design’, in *The 16th Workshop in Primary and Secondary Computing Education*, ACM, Virtual Event Germany, pp. 1–10, doi.org/10.1145/3481312.3481352.

Schätz, E. & Martens, A., 2022, ‘Attributes of physical computing systems – the perspective of pedagogy’, in *Edulearn22 Proceedings*, Palma, Spain, pp. 5561–5565, doi.org/10.21125/edulearn.2022.1309.

Schmalfeldt, Thomas & Przybylla, Mareen, 2021, ‘Erhebungsinstrument für Überzeugungen zum Physical Computing’, doi.org/10.18420/INFOS2021_F205.

Schwarz, Richard, Hellmig, Lutz & Friedrich, Steffen, 2021, ‘Eine Synopse zum Informatikunterricht in Deutschland im Jahr 2020’, doi.org/10.18420/INFOS2021_F227.

Scratch Foundation, 2022, ‘Scratch - Imagine, Program, Share’, in *Scratch*, viewed 18 July 2022, from https://scratch.mit.edu/.

Wilhelm, C., 2021, Feiern Cozmo und Vector bald ein Revival?, viewed 18 July 2022, from https://www.smarthomeassistent.de/feiern-cozmo-und-vector-bald-ein-revival/.