A Categorizing Taxonomy for Occurring Problems During Robotics Activities

Sandra Schulz
Humboldt-Universität zu Berlin
Unter den Linden 6
Berlin, Germany 10099
saschulz@informatik.hu-berlin.de

Niels Pinkwart
Humboldt-Universität zu Berlin
Unter den Linden 6
Berlin, Germany 10099
pinkwart@informatik.hu-berlin.de

ABSTRACT

Robotics and other interactive devices are new forms of learning in computer science education. Currently there is a lot of research going on with respect to appropriate constructions and other device focused investigations. Recent papers describe students' problems as a side product and without a deeper analysis. However, to implement these devices in a learning setting we need to be aware of concrete problems the students are struggling with. The goal of this paper is to give first empirical results concerning the identified research gap. Therefore, we observed students working with two different devices to figure out occurring problems and problem sources students are confronted with. We found hardware, software and environment as three main problem sources. Additionally, difficulties lie in mathematics and physics, which seems to be more an outside problem source. The students seem to have in particular difficulties with ambiguous problems. As a next step we will develop scaffolds to support students' evaluation of the problems and to help the students to categorize the problems they have to tackle.

CCS CONCEPTS

• **Social and professional topics → K-12 education**; • **Computer systems organization → Embedded and cyber-physical systems**; *Robotics*;

KEYWORDS

Problem solving, robotics, computer science education, categorizing problems

ACM Reference format:
Sandra Schulz and Niels Pinkwart. 2017. A Categorizing Taxonomy for Occurring Problems During Robotics Activities. In *Proceedings of WiPSCE '17, Nijmegen, Netherlands, November 8–10, 2017,* 4 pages.
https://doi.org/10.1145/3137065.3137078

1 MOTIVATION

The maker movement is increasing in different age groups and contexts. It is present in shape of numerous devices, uncovering the challenge to support students in their learning process independently of a concrete device. In our work we set the goal to observe problems and problem sources which occur during students' work with robots. In the literature related to robotics and other physical computing (PhC) devices, we found a lot of research in the area of improving student's motivation and skills. We figured out that a description of knowledge domains in this research area is missing. Besides information about what device improves which skill, it would be beneficial to know general problems the students have to tackle. This seems to be necessary to construct classroom settings and to support students' learning.

2 LITERATURE REVIEW

The PhC research area in an educational sense is quite new. Some researchers describe particular devices and classroom settings in this field. Recent results are in nature of enhancing motivation, particularly for girls [4]. In a cross-cutting project other researchers found an increase of confidence to engage with technology in grade 7 and 8 [2]. In addition, building devices with the do-it-yourself approach and to conduct personally relevant products is increasing [8]. O'Sullivan and Igoe describe PhC as "creating a conversation between the physical world and the virtual world of the computer"[7][p. xix]. It includes the components hardware, software and environment, and is also leading to the idea of different problems occurring because of the constructed conversation.

Most papers discus PhC devices without figuring out occurring problems in detail. Initial hints are included in Okita's investigations where the recursive feedback is pointed out [6]. The term "recursive feedback" describes the discrepancy between the written program for the robot (or PhC device) and the following outcomes. When the robot is performing, the programmer has no opportunity to influence the running program and afterwards the student needs to match the outcomes to the accountable parts of program code. Other problems are described by Kafai et al. in an e-textile project [3]. One problem source was to craft functional circuits, in more specific how to connect multiple positive and negative poles. Sewing of the circuits is an other problem. Obviously the program code is a source of problems too. The authors also describe general problem sources like: "However, debugging e-textiles is a complex process, more so than debugging program code, because bugs can be caused by the code, circuit design, or crafting"[3][p. 1:15]. In respect of this, other research motivates to work with hardware modules and a virtual IDE to program the devices. They tried to avoid the overlaps of concepts and necessary knowledge about physics when the goal is to learn computer science [5]. Some approaches show success for the implementation in the classrooms and for young children. Here

---

WiPSCE '17, November 8–10, 2017, Nijmegen, Netherlands

ACM ISBN 978-1-4503-5428-8/17/11…$15.00
https://doi.org/10.1145/3137065.3137078

identified are problems in the software, written by the students, physics knowledge concerning the construction of a correct circuit, and manual skills for crafting. Other studies describe, that students were very attracted by the hands-on nature of robotics projects and surprised about their creative work "by the challenges they found in debugging their programs" [1]. That leads to the software, particularly the written program, as frequent problem source too. In our prior work we analyzed the process of PhC and described some occurring problems [9]. Derived from this investigation and following studies the data of occurring problems is extracted. With respect to the literature we argue, that bringing two diverse worlds together in a PhC system results in more than just the sum of both worlds. Additionally, it seems to be a hurdle to decide what problem is lying in which problem source. Currently in literature there is no systematic analysis regarding problems and problem sources during robotics activities. Nevertheless, it would be valuable to profoundly analyze students' learning in these tasks.

3 CONDUCTED STUDIES

We analyzed data from several conducted studies. To increase the transparency we will give a short overview regarding the design of the studies and explain the methodological approach for the data analysis afterwards. The goal of these studies is to observe students' problems during robotics activities to later conclude with a categorizing taxonomy.

3.1 Study Design

We conducted four similar studies with different groups of students aged between 14-18. Three groups worked with LEGO Mindstorms robots (generation EV3). The first and second of which got no support, the third light support. Light support in this context means, that they where requested to categorize occurring problems they had instantly in problem sources and decided in which source they would make changes to fix the problem. Or in more general, they were motivated to open the black box "robot". They solved one to two tasks, at first to drive around a box with the robot using an ultrasonic sensor. For this task they got approx. 70 min. Fast groups got a second task afterwards, to find the way out of a labyrinth. In these robotics studies overall 15 students participated.

The fourth study was conducted with 10 students working with Arduino microcontroller. They also had one to two tasks to solve and got no additional support. The tasks were to construct a traffic light as output related to the intensity of light measured in the environment. The students got approx. 45 min and for fast students the task to connect a light sensor with a motor and to search for a source of light. The students working with LEGO Mindstorms robots used a block based language from LEGO to program the EV3 robots. For the Arduinos a block based language called *Scratch for Arduino* was used as well. Scratch is also a programming environment most students know from their computer science courses. Most students had prior knowledge in computer science and programming, but mostly without robots. They also participated in an introducing workshop for one and a half day before the studies were conducted.

3.2 Qualitative Analysis

The students were video recorded during the observed tasks. After that the video tapes were transcribed and analyzed with a qualitative content analysis. At first the categories were build deductively from literature and refined inductively based on some cycles of reviewing the transcripts. A problem is coded when 1) the students asked for help, because they could not continue with the task, 2) the teacher decided to help, because the students where struggling for a couple of minutes, or 3) the transcriber of the video tape described a situation where the students made mistakes with/without noticing them. Eventually, the following codes for the transcript analysis were found: hardware (HW), software (SW), environment (EN), mathematic/physics (MP), hardware and software (HW/SW), hardware and environment (HW/EN), software and environment (SW/EN), and system (SY). This differentiation addresses the complexity of the occurring ambiguous problem. It could be just in a single problem source (like HW, SW, EN, MP) or in more dimensions. Two dimensions means, that this problem can lie in both problem sources and can be solved in at least one of which. The last category (SY) contains three dimensions, the overlap of HW, SW and EN - so it is not possible to exclude one of the components of the PhC system.

4 RESULTS

In the following section we present a categorizing taxonomy to explain occurring problems and their sources students have during the process of PhC, derived from the qualitative coding. Afterwards, we look in more detail in these categories, focusing on the overlaps of problem sources or so called ambiguous problems, lying in more than one problem source. To test for inter-coder agreement, we compared the classification of problems to problem sources with the opinion of a second coder who received identified problems had to decide on a category for the problems. All categories were verified, just with slight differences. In some cases the inter-coder chose more categories at the same time in comparison to the first coder.

4.1 Categorization of Main Problem Sources

At first we analyzed occurring unidimensional problems derived from literature. As main categories we found hardware, software and environment - the components of the PhC system. Regarding to the problems found, the main categories can be divided into subcategories, which are shown in table 1. All components of the physical device belong to the *hardware*. The dimension HW contains the levels 1a) the general construction of the device. Sometimes a misconstruction was build and thus a circuit broken. 1b) the sensor is broken or 1c) the misfunction of a sensor. This means the sensor is working but calibrated wrong. An example for the misconstruction is, when students build a robot and connected too long wires in a way that it was impossible to drive ahead, because the wheels ran over the wires. It could also be a wrong connection of wires in a breadboard or robot, which is causing a broken circuit. The *software* contains everything in the system which is programmed and the students interact with. It can occur in the three different levels 2a) the software written by the students, for example a loop or clause was missing, a semantic as well as logical mistake. Some

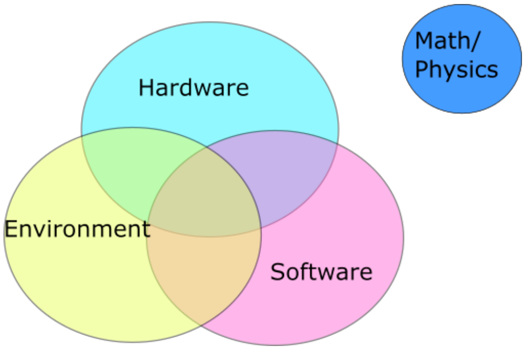

Figure 1: Main categories of problem sources in robotics activities

Table 1: Subcategories of the main problem sources

Main Category	Subcategory
Hardware	Construction (1a) Misfunction of sensor (1b) Broken sensor (1c)
Software	Program code (2a) Programming environment (2b) Firmware from robot (2c)
Environment	Natural environment (3a) Interfere from human (3b)
Math/Physics	use of operators (4a) physical function of sensors (4b) construct a physical experiment (4c) determine a threshold (4d)

students forgot to implement a “forever loop” and the program was finished before the students had the chance to look at the outcomes or that anything can be effected by the PhC system. 2b) problems concerning the programming environment, like uncertainty about the function of program blocks and how to readout the sensor values 2c) or the firmware from the robot. The students struggled for example with picking their intended program or to give the program an appropriate name. The *environment* contains the physical influences on the PhC system and is dichotomous 3a) the natural environment could be changed. Like the light conditions change or are different in a room that effects the PhC system and causes unintended outcomes. Or 3b) there is an interference of a human being, e.g. when changing the light conditions because its covering accidentally the light sensor.

An additional problem source we observed is called mathematics/physics. It contains prior knowledge from science contents, for example concerning a physical experiment. The students had many problems to define a specific threshold (4d) and to decide whether it is better to compare the sensor values with a single value or an interval (4a). Problems we did find in this category were: how to use operators appropriate for the purpose; how the sensor is working (4b); and knowledge about the term and how to handle thresholds. In this taxonomy there is no overlap illustrated with other main or subcategories. Obviously there exist overlaps, like the knowledge how to construct an experiment (4c) is directly linked to manual skills handling the hardware. To show these overlaps a bigger data base would be necessary.

Remarkable but not surprising is, that the students working with Arduinos had more hardware problems than the group with robots. That is most probably the case, because the construction from Arduinos is more complex and the students had more components to change and make more decisions how to change their construction. Here problems occurred like a wire was accidentally put into the wrong row in the breadboard or a wire was not constructed stable enough which leads to a broken circuit. These described mistakes are not possible with some robots, because of their construction and the aspired reduction for cognitive load. Particularly interesting categories are the overlaps of two or more main categories. Then it is called an ambiguous problem and builds a new category. During the analysis we did find, that for much occurring problems it is not possible to decide to which of the main categories they belong. This means the problem is lying in more than one problem source and can be solved through either tackling a problem from the one source or the other (or in both). Because of this, we will describe ambiguous problem sources in the following subsection.

4.2 Ambiguous Problems

The main categories overlap in the derived taxonomy and build four further categories. These are HW/SW, HW/EN, SW/EN and SY. As described in the coding scheme, the categories contain problems, lying in both or all main categories (except PM). Either it is not possible to define in which problem source the problem lies, or to tackle the problem at least addressing one of the problem sources is appropriate to solve the problem. An example is a robot driving around a box just using the ultrasonic sensor in the front of the robot. The gray radiances (see fig. 2) are illustrating the area the sensor is able the measure (and to notice the box). The construction of the robot contains two contact sensors on the left and right front. These sensors are not programmed, because they are not necessary to solve the task but included in the general construction. An occurring problem was that the robot is hitting the box with a contact sensor without recognizing, because they are not programmed and it is not within the limits of the ultrasonic sensor. Because of this, the robot can not follow the intended way or drops. So, the program is not appropriate anymore to solve the task. In our taxonomy this problem is mapped into the category HW/SW. There exist two main possibility spaces to solve this problem. In the HW, it would be possible to change the construction, e.g. to get rid of the contact sensors. The other space is the SW, e.g. to change the program that the robot is keeping more distance to the box. Both possibilities are appropriate solutions in the same way.

An other example addresses the category SW/EN. Most the students put the robot in front of the box, so it needs to drive ahead at first until it is close enough to the box and to turn afterwards right or left. The ultrasonic sensor is constructed above the button to start the program on top of the robot. If the students programmed their robot to turn directly after noticing something in front of the robot, the robot will probably not reach the box, because it needs to drive at first ahead. The problem in this situation is, that the hand in front of the robot when starting the program was measured and

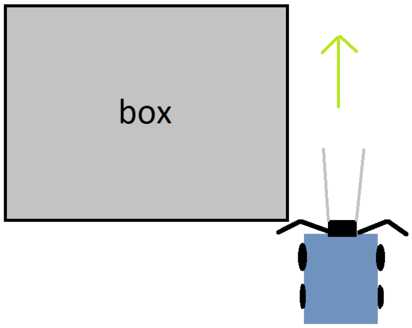

Figure 2: An ambiguous problem between hardware and software

not (like intended) the box. One solution would be to change the construction and to put the sensor below the button, or to program the software not to measure at the beginning and drive ahead at first for one second.

An ambiguous problem in HW/EN occurred, when the students programmed the traffic light with an Arduino depending on the intensity of light. Some students constructed the green bulb (for high intensity) close to the light sensor. When they illuminated the sensor with a flashlight to test if the program is working, they also illuminated the green bulb and they could not see that it is turned on. The flashlight was too intensive and the bulb to weak. That is why the students thought their program was not working and started to make changes.

The problem sources SY is explained in the following example. To drive straight is one component to solve some of the tasks. Students struggled with that problem, because it is effected by all the components of the PhC system. A dusty ground can cause the problem and the wheels are slipping. It is also possible that one motor is running faster than the other motor. Or the robot is heavier on one side and the result is a curve. Yet, mistakes in the program or an adjustment in the SW are possible as well. The category MP (mathematics and physics) is in the taxonomy standing back. There exist overlaps to HW, SW and EN as well, but this is not focused in our research.

Surprising is that many problems occurred in these ambiguous problem sources. It seems to be the biggest challenge during the work with PhC devices because it is hard to determine and not familiar from computer science lessons. The most systems in computer science we usually work with do not contain the dimensions hardware and environment.

5 CONCLUSION

This paper bridged the gap between PhC as a tool and the development of support for students. We did find different problem sources which can be categorized in *hardware*, *software* and *environment* based on the definition of PhC. However, using inductive categories we found more delicate subcategory and a further category. These are the overlaps of main categories and the additional category *mathematics/physics*. This taxonomy is a starting point to give instructions for teachers and students to support the problem solving process during PhC activities. In respect to these results, other research in PhC can build on the taxonomy, e.g. to educationally design new devices or software tools.

Building the taxonomy we had a limited sample in used devices, tasks and age groups. Therefore we tried to stay quite general in the categories, to make them transferable to other PhC devices and tasks. Further considerations of the limits will be tackled in our future work and are described in section 6.

6 FUTURE WORK

Our future goals are twofold. On the one hand the taxonomy needs to be validated with more data. This includes a bigger variety of robotics even PhC devices and tasks. For that a differentiation into more dimensions (like misconceptions, a lack in skills or knowledge) seems to be appropriate. In our analysis we divided the problems in these categories, yet there is not enough data for a deeper system of categorization. On the other hand appropriate ways to support students need to be carried out. We already derive a few hints from the occurred problems, which would have been helpful for the students in this situation. Afterwards we can generalize this bunch of hints into a scaffold, aiming to open the black box of a PhC system. As a first approach we will implement a scaffold in a classroom setting and analyze the students' learning outcomes in a pre- and post-test. Therefore we will use a self-constructed test addressing the identification of problems and components of a PhC system, inspired by Sullivan [10]. Additionally, we will measure the gain of programming abilities and conduct interviews. We are aware that this can only provide initial hints about the success of the scaffold. In respect to the results we aim to reconstruct the scaffold for a classroom use and for teachers.

REFERENCES

[1] Jennifer Cross, Emily Hamner, Lauren Zito, and Illah Nourbakhsh. 2016. Engineering and computational thinking talent in middle school students: a framework for defining and recognizing student affinities. In *Frontiers in Education Conference (FIE), 2016 IEEE*. IEEE, 1–9.

[2] Jennifer L Cross, Emily Hamner, Chris Bartley, and Illah Nourbakhsh. 2015. Arts & Bots: application and outcomes of a secondary school robotics program. In *Frontiers in Education Conference (FIE), 2015. 32614 2015. IEEE*. IEEE, 1–9.

[3] Yasmin B Kafai, Eunkyoung Lee, Kristin Searle, Deborah Fields, Eliot Kaplan, and Debora Lui. 2014. A crafts-oriented approach to computing in high school: Introducing computational concepts, practices, and perspectives with electronic textiles. *ACM Transactions on Computing Education (TOCE)* 14, 1 (2014), 1.

[4] Fatima Kaloti-Hallak, Michal Armoni, and Mordechai Moti Ben-Ari. 2015. Students' Attitudes and Motivation During Robotics Activities. In *Proceedings of the Workshop in Primary and Secondary Computing Education*. ACM, 102–110.

[5] Eva-Sophie Katterfeldt, David Cuartielles, Daniel Spikol, and Nils Ehrenberg. 2016. Talkoo: A new paradigm for physical computing at school. In *Proceedings of the The 15th International Conference on Interaction Design and Children*. ACM, 512–517.

[6] Sandra Y Okita. 2014. The relative merits of transparency: Investigating situations that support the use of robotics in developing student learning adaptability across virtual and physical computing platforms. *British Journal of Educational Technology* 45, 5 (2014), 844–862.

[7] Dan O'Sullivan and Tom Igoe. 2004. *Physical computing: sensing and controlling the physical world with computers*. Course Technology Press.

[8] Mitchel Resnick and Eric Rosenbaum. 2013. Designing for tinkerability. *Design, make, play: Growing the next generation of STEM innovators* (2013), 163–181.

[9] S. Schulz and N. Pinkwart. 2016. Towards Supporting Scientific Inquiry in Computer Science Education. In *Proceedings of the 11th Workshop in Primary and Secondary Computing Education (WiPSCE '16)*. ACM, New York, NY, USA, 45–53. https://doi.org/10.1145/2978249.2978255

[10] Florence R Sullivan. 2008. Robotics and science literacy: Thinking skills, science process skills and systems understanding. *Journal of Research in Science Teaching* 45, 3 (2008), 373–394.