Physical Computing in STEM Education

Sandra Schulz
Humboldt-Universität zu Berlin
Department of Computer Science
saschulz@informatik.hu-berlin.de

Niels Pinkwart
Humboldt-Universität zu Berlin
Department of Computer Science
pinkwart@informatik.hu-berlin.de

ABSTRACT

STEM education research has received considerable attention in recent years. The goal of STEM education is to achieve a common literacy across different STEM subjects and to strengthen every single subject via the combination with other subjects. We present a way of incorporating Computer Science elements in STEM education and a small case study which tests the suitability of this approach, finding initial positive evidence.

CCS Concepts

•**Computer systems organization** → **Embedded systems;** Robotics;

Keywords

Computer Science Education, Physical Computing, STEM Education

1. INTRODUCTION

The widely spread word "STEM" is an acronym for science, technology, engineering and mathematics. Yet, while the term definition is agreed upon, the exact meaning and intention behind STEM (and STEM education) is less clear. In literature, we often find STEM in the contexts of natural sciences and mathematics, much less in the contexts of engineering and technology. Yet, the process of problem-solving in the 21st century is closely linked to engineering and technology [2], and particularly requires competences in computer science (CS). In current STEM education, though, it seems that many teachers do not integrate ideas of computer science much beyond the pure use of computers as a tool - even though the importance of basic education in computer science is indisputable. The goal of this paper is to propose physical computing as a suitable approach for incorporating computer science in STEM education. We present the results of a small case study as initial evidence.

WiPSCE '15 November 09-11, 2015, London, United Kingdom

ACM ISBN 978-1-4503-3753-3/15/11.

DOI: http://dx.doi.org/10.1145/2818314.2818327

2. PHYSICAL COMPUTING IN STEM

In order to integrate CS education with education in other STEM subjects more closely, approaches that combine digital elements with the "real" world seem suitable. Physical computing seems like a suitable candidate here as it "is about creating a conversation between the physical world and the virtual world of the computer" [5][p. xix]. It is the chance to conduct interesting educational projects even with relatively low technical experience. One reason for the success of prior studies could be that physical computing is a form of CS which is connected to arts – which leaves a lot of room for creative work in the classroom. Additionally, physical computing offers diverse connections to other STEM subjects, such as the simulation of behaviour (biology), the collection and analysis of measurements (physics), and logical operations (mathematics).

2.1 Case Study

In May 2015 we conducted a study with student teachers at Humboldt-Universität zu Berlin. The goal of the study was to determine the suitability of a specific physical computing example lesson for STEM education, explicitly including CS education as one component. The rationale behind the study was to have student teachers (inexperienced with physical computing but knowledgeable in STEM) go through some typical lesson elements of a physical computing STEM course, and let them comment of the suitability of the approach afterwards.

Nine students participated in the study. All but one of them had completed courses on CS education, and all had teaching experience in schools, encompassing one to three internships, each of four weeks. All student teachers had CS as one study field. The second subject of study of the participants included mathematics, physics, chemistry or history. The participants had little to no experience using LEGO Mindstorms robots and were not familiar with its programming interface. They had however completed one to two University courses on the principles of computer programming.

In the study, the participants were first introduced to the goals and procedure of the study. Afterwards, the field of physical computing, the programming interface (we chose to use the LEGO Mindstorms software with a modular design), the construction, sensors and actuators of the robot, were described to the participants. Next, the participants received one robot of the EV3 generation each. After this, the participants got some to should be solved with the robots. The tasks included two interdisciplinary tasks about uncertainties of measurements, combining physics and CS (suitable for grade 9 to 12). The participants explored the capabilities and limits of the ultrasonic sensor and applied their results to programming tasks, for instance to pass through a labyrinth. Then, they were asked to conduct tasks that connect biology and CS by simulating individual and group behaviour. Here, the participants used the Bluetooth connection between the robots to establish communication. After the tasks were handed out, the working process was not interrupted. Finally, the students got questionnaires that contained established competence lists for the grades 9-12 from the ACM [3], the NGSS (biology) [1], and Gott et al. (physics) [4]. The participant's task was to select the competences from the lists of which they believed that the lesson elements related to. We used a scale with three answer options: (1) the competence can not be improved with reasonable effort with this physical computing lesson, (2) the competence can likely be improved with this physical computing lesson, or (3) the competence can be improved very well with this physical computing lesson.

2.2 Results

Table 1 shows the results of the questionnaire in a short form. The scales are classified in computer science, physics and biology. **CS** competences are divided into five subscales. The percentage numbers indicate how many of the students believed that the competences associated to this sub-scale can be improved with physical computing (i.e., they answered with option 2 or 3 in the questionnaire). If the table contains a range and not a single percentage for a sub-scale, this means that the answers to the questions varied in the sub-scale.

The first CS sub-scale is *computational thinking*. Here, all participants agreed that competences "explain how sequence, selection, iteration, and recursion are building blocks of algorithms" and "use modelling and simulation to represent and understand natural phenomena" can likely or very well be supported with physical computing. The items "describe a software development process used to solve software problems" and "discuss the value of abstraction to manage problem complexity" were classified by 78 % as suitable or very suitable for physical computing, and "describe how computation shares features with art and music by translating human intention into an artefact" were chosen by 89 %. "Compare techniques for analyzing massive data collections" and "describe how various types of data are stored in a computer system" were less consented with 11 %. In the sub-scale *collaboration*, all student teachers agreed that the development of software in a team can be encouraged via physical computing lessons. Some competences in the sub-scale *computing practice and programming* were highly assessed with 78-89 % agreement: "use various debugging at testing methods to ensure program correctness" and "apply analysis, design, and implementation techniques to solve problems (e.g. use one or more software life-cycle models)". The fourth sub-scale *computers and communication devices* was completely highly rated with 89-100 % agreement for encouraging embedded systems, principals of computer organization and robotics. The last sub-scale on *Community, global, and ethical impacts* received less agreement.

Competences in the field of **physics** were analysed in the sub-scale *calibration and error*. For more than half of the items in this sub-scale, all participants uniformly stated that the corresponding competences can likely or very well be improved with Physical Computing. These competences are "intervening points", "sensitivity", "resolution and error", "instrument use" and "human error". For the other the items, between 78-89 % of the participants agreed that the related competences (end points, zero errors, limiting sensitivity and specificity) can be supported.

Scale	Subscale	%
CS	Computational Thinking	78-100
	Collaboration	100
	Computing Practice & Programming	44-89
	Computers & Communication Devices	89-100
	Community, Global & Ethical Impacts	11-78
Physics	Calibration & Error	78-100
Biology	Role of group Behavior	78-100

Table 1: Questionnaire results

In the field of **biology**, there is one sub-scale on "evaluate the evidence for the role of group behavior on individuals and species‘ chances to survive and reproduce". Nearly all participants (78-100 %) believed that following competences can likely or very well be improved with physical computing: "distinguishing between group and individual behavior", "identifying evidence supporting the outcomes of group behavior" and "developing logical and reasonable arguments based on evidence. Examples of group behaviors could include flocking, schooling, herding, etc.".

3. DISCUSSION AND CONCLUSION

In this paper, we proposed physical computing as a candidate field for CS affine STEM education. We discussed the suitability of the approach and presented a case study with student teachers which gave us some initial confirmation for the hypothesis that physical computing can be used to teach CS in conjunction with other STEM subjects.

Our small-scale pilot study can certainly only be a first starting point of a longer investigation on incorporating CS more closely into STEM education. As a next step, we plan to conduct semi-structured interviews with students and teachers, because our study participants gave the feedback that the competence lists we used in our study would be worth discussion. Having completed that, we will develop more course sequences and conduct empirical studies to test these sequences. Our long term goal will be to create an empirically validated STEM curriculum that pays specific attention to CS education.

4. REFERENCES

[1] Achieve, Inc. Next Generation Science Standard.

[2] Rodger W. Bybee. What is stem education? *http://www.sciencemag.org/content/329/5995/996.full*, 2010.

[3] Computer Science Teachers Association. CSTA K-12 Computer Science Standards, 2011.

[4] Richard Gott, Sandra Duggan, Ros Roberts, and Ahmed Hussain. Research into understanding scientific evidence. *http://www.dur.ac.uk/rosalyn.roberts/Evidence/cofev.htm*, 2003.

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