International Society of the Learning Sciences

The Invisibility Issue: High School Students' Informal Conceptions of Everyday Physical Computing Systems

Gayithri Jayathirtha, Yasmin Kafai
gayithri@upenn.edu, kafai@upenn.edu
University of Pennsylvania

Abstract: While making physical computational artifacts such as robots or electronic textiles is growing in popularity in CS education, little is known about student informal conceptions of these systems. To study this, we video-recorded think-aloud sessions (~10 minutes each) of 22 novice CS high school students explaining their understanding of everyday physical computing systems and qualitatively analyzed transcripts and student drawings for their structural, behavioral, and functional understanding of these systems. Most students identified the presence of programs in making these systems functional but struggled to account them structurally and behaviorally. A few students pointed out probable programming constructs in shaping underlying mechanisms, drawing from their prior programming experiences. To integrate these systems in computing education, we call for pedagogical designs to address the invisibility of computation—both of structural interconnections and of program execution.

Introduction

Physical computing systems (PCS) which involve microcontrollers, inputs, and actuators are used for designing art installations and robotic systems that can interact with the world (O'Sullivan & Igoe, 2004). As computing education grows in K-12 classrooms, these systems are being adopted across a variety of learning contexts. Novices' extensive user experience with everyday PCS may provide productive informal conceptions about these systems—what constitutes these systems and the mechanisms to realize observed outcomes. Indeed, one hope in using PCS in education is that in constructing these systems, learners will draw from their experiences as users, thereby transforming computing learning from being abstract to relevant and concrete (Przybylla & Romeike, 2014). However, relatively few studies have explored how novice learners interpret everyday PCS.

Physical computing systems traditionally consist of input (e.g., buttons and sensors) and output devices (e.g., actuators such as lights, speakers, motors) connected to a central processing unit. The processing unit receives input signals, performs required computation based on a program, and sends appropriate signals to the output devices (e.g., O'Sullivan & Igoe, 2004). PCS can be viewed as consisting of three key dimensions: structure, behavior, and function (SBF) (Bhatta & Goel, 1997). The physical parts of the system make the structural composition, while its overall purpose in terms of inputs and outputs covers the functional aspects; and, the underlying logic that brings together the different components in action accounts for the behavior (Bhatta & Goel, 1997). Extending this framework to understand PCS such as sensor-based stoplights and interactive toys: (1) the *structural* aspects include the different components such as sensors, lights, and the physical connections; (2) the *functional* aspects consist of the roles of each of these components and the overall goal; and (3) the *behavioral* aspects include the underlying logic causing the functionality.

Prior studies around novices' informal conceptions of PCS have touched upon certain aspects of SBF although they shed limited light on students' understanding of the role of computation. For instance, Cederqvist (2020) examined middle schoolers' conceptions of PCS such as car remote key to uncover their structural and functional understanding and found that most of their understanding pivoted around the visible components such as buttons and lights. Pancratz and Diethelm (2020) observed a range of ways in which middle and high school students understood the structural aspects of daily-use PCS such as robotic vacuum cleaners and video game consoles. But, this analysis—limited to uncovering students' structural understanding—revealed very little about how they understood the functional and behavioral aspects of the systems. With PCS making a foray into introductory computing programs, there is a need to examine students' informal conceptions of these everyday PCS, specifically their understanding of computation in shaping their function and behavior.

In this exploratory study, we conducted think-aloud interviews (Ericsson & Simon, 1998) with 22 high schoolers enrolled in an introductory computing class. We provided pictures of two common PCS, a sensor-based stoplight and an interactive plush toy, and invited students to explain SBF aspects of each. In this paper, we answer the following questions: (1) What aspects of computation did students attend to in their descriptions? And, (2) How did these descriptions qualitatively differ across students?

Methods

Context and participants

The study was conducted at a public charter high school located in a large U.S. west coast city. Students were in a year-long introductory computing course, *Exploring Computer Science* (Goode, Chapman, & Margolis, 2012). The class had 38 students, 22 of whom the teacher chose to participate in interviews (14-18 years; 11 female, 11 male; 8 White, 6 Latino/Hispanic, 6 Asian, and 2 African-American). At the time of the interview, they had 10 weeks of Scratch programming experience and this was the only programming experience for most of them.

Data collection and analysis

The first author conducted and video-recorded semi-structured think-aloud interviews (Ericsson & Simon, 1998) where every student was asked to draw and explain their understanding of two different PCS: a sensor-based stoplight and an interactive toy (see Fig. 1)—the former present within a mile radius of the school and the latter, an artifact that students can relate to. Each interview lasted about 10 minutes.

The first author annotated the interview transcripts to capture student gestures and gaze, and drawings to capture related utterances. These were analyzed initially deductively and later inductively (Creswell & Poth, 2016). Deductive analysis involved categorizing descriptions for SBF. Another researcher independently coded 3 student transcripts and drawings (~10% of data), reaching ~90% agreement. After discussing disagreements, the revised codebook was applied, and themes were generated to capture qualitatively differences among descriptions: *simplistic* descriptions that discounted any computation or automated control; *qualified* descriptions that only identified a control mechanism without elaborating on its nature or details, and *constructive* descriptions that involved details about computation such as specific programming constructs. Another researcher then coded 10% of the student responses with the new coding scheme, reaching ~85% agreement. Disagreements were discussed, and the first author coded all the student responses. Each student explanation of every PCS was considered as a unit of analysis in which SBF aspects were identified and a qualitative descriptor was assigned to each, resulting in 6 codes for each student across the two interview artifacts (6 circles per student in Fig. 2).

Figure 1. Interview artifacts: Sensor-based stoplight (left) and an interactive toy bird (right).

Findings

Qualitative variations within student explanations and relationships between SBF aspects revealed three student groups: one with qualified functional descriptions for at least one of the two PCS (n=15; middle group in Fig. 2); another with constructive articulation of computational unit structurally and functionally, and specific programming-related details within behavioral descriptions (n=5; left-most group in Fig. 2); yet another group with simplistic explanations (n=6; right-most group and Jade in Fig. 2).

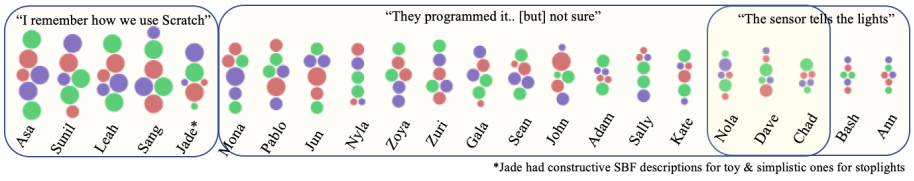

Figure 2. Visualization of student descriptions; red, blue, and green for structural, behavioral, and functional descriptions respectively; small, medium, and large circles for simplified, qualified, and constructive descriptions respectively (all student names are pseudonyms).

"They programmed it…[but] not sure:" Qualified functional descriptions

More than half of the students identified computation in their functional descriptions. They listed an automated mechanism that shaped the overall system functionality (n=15; students with at least one medium green circle, mostly in the middle group in Fig. 2). However, these students struggled to articulate the role of computation in the underlying behavior (medium or small blue circles in Fig. 2). Students identified components such as lights and audio devices as outputs, and the sensors, buttons, and cameras as input devices while describing the system's

functionality; most of them (n=9) extended their structural descriptions to acknowledge some form of a computational unit in organizing these inputs and outputs (medium or large red circles in Fig. 2). For instance, Sally identified the control mechanism as a "machine behind [the stoplight]" to coordinate sensors and lights while others acknowledged the presence of a "motherboard," "chip," or a "control panel" structurally.

However, identifying the presence of a computational unit structurally or acknowledging its role functionally did not mean they connected it to system behavior. Many students in this group (first 8 of 13 in the middle group in Fig. 2), informed by their prior Scratch programming experience, thought of computation as distributed across different components such as input/output devices, similar to how sprites are associated with code fragments in Scratch (Maloney et al., 2010). This was evident in Zuri's descriptions of programs as residing in the toy's sensors that will control signals sent to the lights (see Fig. 3, middle). Others (last 5 of 13 in the middle group in Fig. 2) slipped back to simplistic descriptions, barely acknowledging any computation. For instance, Sally, who initially noted the presence of a computational machine controlling the system, dropped it in her behavioral description, and told that the sensors and the lights directly communicate with each other. Overall, students' struggles with articulating the role of computation in system behavior despite identifying its functional presence points to the accessibility of the functional compared to the behavioral or even structural aspects.

"The sensor tells the lights:" Simplistic SBF descriptions

A number of students missed accounting for any computation across their SBF descriptions for at least one of the two PCS (6 out of 22; right-most group and Jade in Fig. 2). The perceptual aspects of the artifacts were the most accessible for learners as they anchored their descriptions around visible inputs (e.g., sensors or buttons) and outputs (e.g., lights and speakers) while leaving out any processing unit. For instance, Bash drew (Fig. 3, left) buttons, a "tape recorder," lights, and wires, and directly connected them to each other without any computing unit. Simplistic structural descriptions led to simplistic functional explanations. In this case, Bash continued to explain buttons as simple electronic switches directly controlling the lights and the "tape recorder." leaving no room for any explanation related to its conditional outcomes based on different degrees of presses.

Simplistic structural and functional descriptions further led to simplistic behavioral explanations. Some students adjusted their simple circuits connecting inputs and outputs to make up for computation-based sensing functionality. For instance, Nola, similar to Bash, thought of the toy in simplistic terms as consisting of buttons, lights, and a microphone connected through wires. When asked about the conditional outcomes, she included computation within her simplistic structural descriptions by adding a set of wires between the buttons and the lights that could send "two different types of signals." She continued to explain the mechanism as one where the button allows for two different kinds of presses and sends different sets of signals for the lights to display patterns. Overall, visible aspects of the PCS such as input and output devices were most accessible to these students, similar to Cederqvist's (2020) observation, and this limited students' ability to see interconnections and computation across the structural, functional, and behavioral aspects of the systems.

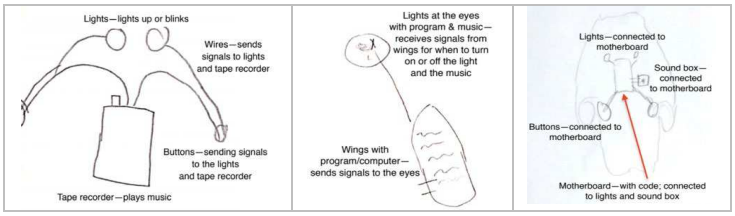

Figure 3. Annotated student drawings: Bash's simplistic connections between the buttons, lights and "tape recorder" (left); Zuri's qualified drawing of wings with computing abilities (middle); Sang's constructive drawing showing "motherboard" connecting the inputs and outputs (right).

"I remember how we use Scratch:" Constructive SBF descriptions

Although most students faced challenges in either identifying or integrating computation into their descriptions, a small proportion of them (5 of 22 in the left-most group in Fig. 2) described specific aspects of computation. They acknowledged the presence of a central processing unit in their drawings and explained its function as receiving inputs and sending signals to respective devices. Further, they drew from their limited prior programming experience to predict probable programming constructs controlling the system behavior. As seen above, Sang's explanation of the "motherboard" connecting the lights, "sound box," and the buttons in the toy bird clarified the connections between microcontroller-like processing, input, and output devices and supported his further functional and behavioral explanations (see Fig. 3, right). He connected the sensor-based interactive functionality to programming constructs by including *if-else* statements to explain conditional outcomes.

Similar to Sang, most of his peers elicited their explanations from their recent and yet limited Scratch programming experience: pointing to *if-else*, *delay*, and *forever* blocks while explaining the underlying behavior. For example, Asa employed these constructs to explain how stoplights work. "You do make a loop where certain things happen then certain activities occur… someone presses the button on the crosswalk, the green light will go a little bit faster," she said describing the loop and conditional constructs mediating interactions between human users and stoplights. Others identified the *forever* loop block in Scratch (e.g., Jade), the *delay* block to count down time for the lights (e.g., Leah), or the *if-else* conditional blocks to manage the input signals and affect outputs appropriately (e.g., Sunil). Though these students' descriptions were relatively advanced compared to their peers, they were limited too in further elaborating on the data and control flow between the computational unit, inputs, and outputs to account for the overall system behavior. In sum, students' initial articulation of different aspects of PCS promises its potential integration with computing teaching while pointing at pedagogical designs to meet where students are to relate computation to everyday PCS.

Discussion

Contexts such as stoplights have a potential to draw from students' user experience. At a functional level, they allow students to make connections to microprocessor-based learning systems such as Arduino construction kits. However, these user-facing PCS intentionally black-box complex underlying connections and this stands in the way of effective utilization of these systems for educational purposes. One way to get around this is to explore the affordances of student-generated drawings in revealing certain underlying aspects. As noted above, student drawings were effective in surfacing structural understanding and mediating functional and behavioral explanations when combined with the "what" and the "how" questions. Teachers can adopt similar practices to highlight the invisible processing unit and the interconnections and facilitate discussions about systems' function and behavior. To surface the invisible program execution that controls the system behavior, teachers can employ notional machines i.e., pedagogical devices to communicate program execution (Sorva, 2013) and be informed by students' prior programming experiences (e.g., Scratch in the examples above). Future research can explore learning activities to integrate everyday PCS with computing education. To conclude, future research, as well as careful pedagogical design can make room for everyday PCS within computing education.

References

Bhatta, S. R., & Goel, A. (1997). Learning generic mechanisms for innovative strategies in adaptive design. *The Journal of the Learning Sciences*, *6*(4), 367-396.

Cederqvist, A. M. (2020). Pupils' ways of understanding programmed technological solutions when analysing structure and function. *Education and Information Technologies*, *25*(2), 1039-1065.

Creswell, J. W., & Poth, C. N. (2016). *Qualitative inquiry and research design: Choosing among five approaches*. Sage publications.

Ericsson, K. A., & Simon, H. A. (1998). How to study thinking in everyday life: Contrasting think-aloud protocols with descriptions and explanations of thinking. *Mind, Culture, and Activity*, *5*(3), 178-186.

Goode, J., Chapman, G., & Margolis, J. (2012). Beyond curriculum: the exploring computer science program. *ACM Inroads*, *3*(2), 47-53.

Maloney, J., Resnick, M., Rusk, N., Silverman, B., & Eastmond, E. (2010). The scratch programming language and environment. *ACM Transactions on Computing Education (TOCE)*, *10*(4), 1-15.

O'Sullivan, D., & Igoe, T. (2004). *Physical computing: sensing and controlling the physical world with computers*. Course Technology Press.

Pancratz, N., & Diethelm, I. (2020). "Draw us how smartphones, video gaming consoles, and robotic vacuum cleaners look like from the inside" students' conceptions of computing system architecture. In Workshop in Primary and Secondary Computing Education, 2020, Virtual Event, Germany. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3421590.3421600.

Przybylla, M., & Romeike, R. (2014). Physical Computing and Its Scope--Towards a Constructionist Computer Science Curriculum with Physical Computing. *Informatics in Education*, *13*(2), 241-254.

Sorva, J. (2013). Notional machines and introductory programming education. *TOCE*, *13*(2), 1-31.

Acknowledgments

We would like to thank Deborah Fields and the PennGSE RAC team for their generous feedback, Luis Morales-Navarro and Katie Cunningham for help with data analysis, and participating teacher and students for their time. This project was supported by National Science Foundation (#1742140).