International Society of the Learning Sciences

"Do I need to know what I am doing if I am the teacher?" Developing teachers' debugging pedagogies with physical computing

Colin Hennessy Elliott, University of Colorado, colin.hennessyelliott@colorado.edu
Jessie Nixon, Utah State University, jessie.nixon@usu.edu
Jeffrey B. Bush, University of Colorado, Jeffrey.Bush@colorado.edu
Alexandra Gendreau Chakarov, San Jose State University, alexandra.chakarov@sjsu.edu
Mimi Recker, Utah State University, mimi.recker@usu.edu

Abstract. This paper presents findings from a study of middle school science teachers' professional learning activities designed to support the development of their debugging pedagogies. In two iterations of a professional learning activity, teachers worked to find bugs planted by facilitators in physical computing systems they were learning to integrate into their middle school science classrooms. We examine how teachers navigated the tension between developing their own troubleshooting skills versus supporting students' skills in resolving inconsistencies between what they expect of the DaSH and what it actually does. We conclude with implications for the design of PL activities for supporting teachers' debugging pedagogies.

Introduction

There is a concomitant need for designing instructional approaches and tools that integrate computational and data practices into middle school science. As part of a multi-year research practice partnership, our team iteratively co-designed a sensor-based physical computing system, the Data Sensor Hub (DaSH), and accompanying curricular units. The DaSH integrates low-cost programmable sensor technologies to support students in a range of computational activities with data (Gendreau Chakarov, et al., 2021). In the accompanying units, students encounter problems with the DaSH, or bugs, and need to troubleshoot, or debug. For moments of material resistance (Manz, 2015) when the DaSH is not operating as students expect, teachers require pedagogical skills to help students resolve their observed issues and develop their own debugging approaches for future work.

This paper explores how teachers, most of whom do not feel comfortable programming themselves, develop pedagogies to support students to expand their debugging skills. We examine how teachers navigated the tension between developing their own debugging skills and supporting students' learning by analyzing two iterations of a professional learning (PL) activity. In the activity teachers worked collaboratively to find bugs planted by facilitators that commonly occurred in prior classroom implementations. Analyses were guided by the following research questions: RQ1) How did teachers approach debugging in two iterations of the PL activities? RQ2) To what extent did the PL activities foster teachers' evolving debugging pedagogies?

Conceptual Framework: Learning to teach debugging as a situated inquiry

We characterize debugging a physical computing system like the DaSH as a situated inquiry, which "cannot be reduced to pre-specified plans or generalizable procedures" (Flood, et al., 2018; p.1405). Bugs are interactionally produced through a public recognition of an inconsistency between a working expectation of the DaSH and the DaSH's output. Debugging, therefore, requires constructing an expectation, interpreting the DaSH's output, identifying the inconsistencies between the two, and taking action with the system. Debuggers construct expectations of the DaSH by layering inscriptions (Latour, 1986) of the physical system and representations of the environment it interacts with. Inscriptions are physical representations (e.g. charts, graphs, labels) of a process (or physical phenomenon) that social actors use and remake to communicate developing knowledge and associations. The computer program is an inscription of the DaSH's operation while wiring diagrams are inscriptions of the physical setup of a working DaSH. Facilitating students learning to debug as a situated inquiry requires that teachers have flexible knowledge of the physical computing system as well as pedagogical practices. These include teaching moves of modeling and supporting thinking strategies, questioning agendas, and reflecting on applied strategies (DeLeima, et a., 2019).

Methods, Data Sources and Analysis

Teachers in two school districts participated in ongoing PL activities toward implementing instructional units in which students ask questions about environmental phenomena and use the DaSH to measure and visualize data to investigate them. The DaSH is a physical computing system consisting of a micro:bit (microcontroller), a gator:bit, and several physical environment sensors which can be programmed using a block-based programming language to collect and display data using sound, an array of LEDs on the micro:bit, and a row of five LEDs on the gator:bit.

This study examines two iterations of a PL activity where teachers worked together to find bugs that students commonly encountered (columns 2 and 3 in Table 1). In the PL, participating teachers worked in groups to find and fix systems with planted bugs (building on Fields et al., 2021) and discussed how to support students. This builds on a "student hat – teacher hat" approach where teachers experience an activity as students would and then consider the teacher perspective.

Analysis centers on a group from both iterations: Episode A involved three participants (Brett, David, and Grace) from Iteration 1. Episode B involved two participants, Aaron and Chris from Iteration 2. Both groups were given the same planted bugs. The activity was designed for participants using the soil moisture sensor to get the DaSH to display a smiley face when the moisture level was ***high***, and an 'X' when the level was ***low.*** Participants were given a DaSH with two bugs: the wrong program was downloaded onto the micro:bit and the correct program listed the soil moisture sensor's connection to the gator:bit on the wrong PIN.

Table 1.
Overview of debugging PL from two iterations and a proposed third iteration detailed in Conclusion

PL Objectives	Participants practice, observe, and discuss how to facilitate debugging moments that students encounter with the DaSH during the Sensor Immersion Unit.		
	PL Iteration I	**PL Iteration II**	**Future PL Iteration III**
PL method, teacher role	Teacher & student hat	Student hat	*Round 1:* Student hat *Round 2:* Both, mostly teacher hat
PL method, participation	Participants rotate through roles as teacher and students pair programming (driver & navigator)	Participants debug together, no official pair programming roles	*Round 1:* Pair programming *Round 2:* Pair in teacher hat supporting the third
PL group size	Groups of three	Groups of two	*Round 1:* Groups of two *Round 2:* Groups of three
Post PL reflection	Whole group discussion focused on debugging with students	Brief whole group discussion that mainly highlighted what bugs people encountered.	*Round 1:* Reflect on debugging strategies *Round 2:* Reflect on debugging pedagogy
Timing	30 minutes of debugging 20 minutes of discussion	20 minutes of debugging 10 minutes of discussion	*Round 1:* 30 minutes *Round 2:* 60 minutes
Directions provided to participants	Teacher hat: One participant in role of teacher Student hat: driver and navigator pair programming	Given expectation for how program should work, told to fix the bugs	*Round 1:* Given notes about the program's expected behavior, told to fix bugs *Round 2*: Help teach debugging skills
Context provided	Participants were told planted bugs mirrored bugs students have encountered	Bugs were explicitly linked to specific lessons in the Sensor Immersion Unit	Bugs explicitly linked to specific lessons in the Sensor Immersion Unit
Challenges of debugging activity	Participants were confused about how those in teacher hat could help. Three minutes in, facilitators explained bugs to those in teacher hat	Because of time constraints, participants did not have the chance to move into teacher hat or adequately reflect on debugging pedagogy.	TBD: activity planned but not yet implemented

Table 1 shows PL activity structure in each iteration. After the first, researchers recognized that combining teacher and student roles created a complexity that would be better placed after all participants could participate as a student (see Episode A below). Researchers also found that providing participants with a written expectation of what the program was supposed to do (referred to as "Expectation" below) would create direction for debugging activities. Additional shifts were also required due to time constraints for PL in District 2.

We used two qualitative methods of analysis to address our research questions: interaction analysis (IA; Jordan & Henderson, 1995) and a grounded approach with inductive coding. To address RQ1, we analyzed video of teachers' debugging tracing the trajectories of inscriptions of the environment, the DaSH, and other resources. We selected a few episodes to view, following the inscriptions teachers developed, used, and rejected to uncover their approaches to debugging. To address RQ2, we used an inductive coding scheme on the PL activity, whole group discussions, and supplementary data (interviews and surveys) to trace the representations of the role of the teacher. Focusing on where these representations unfolded, and where they did not, we developed a view of how the PL activities supported teachers in developing debugging pedagogies, which we address in the discussion.

Analysis

Episode A: Iteration 1 (Grace, David, Brett)

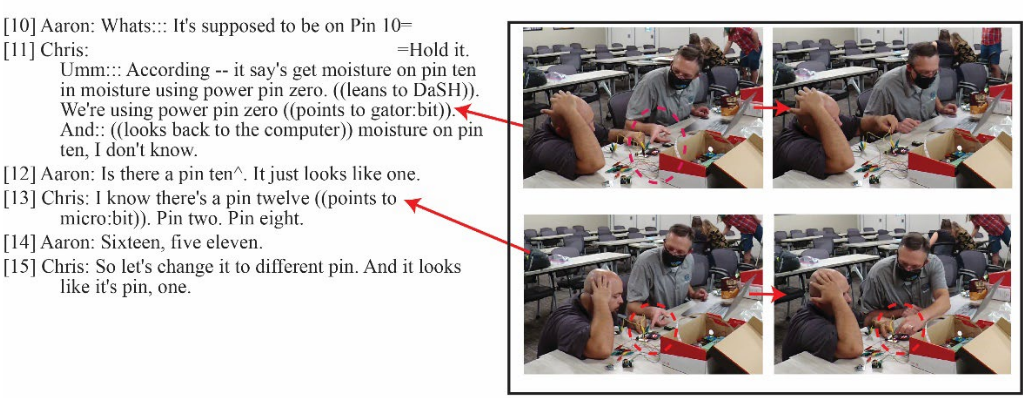
In 30-minutes, Grace, David, and Brett they got the DaSH to perform almost as expected by debugging as a group. At the start, Brett organized them into prescribed roles appointing David the navigator, Grace the teacher (which she initially balked at), and himself the driver. Thereafter, Brett was the only one to physically interact with the DaSH or change the program, limiting Grace and David's access to related and important inscriptions. Brett's interactions consolidated his access to the DaSH and therefore his role as the driver. David's moves supplanted his navigator role as a support for Brett's investigation by, for example, offering the wiring diagram as a useful inscription. When approached by the facilitator to show her the planted bugs, Grace shared: "*Do I have to know what I'm doing to be the teacher?*" After hearing what the bugs were, she showed a new confidence, saying "*I know the answer.*" Knowing "the answer" dramatically shifted Grace's contributions. Before, she was tentative. Afterwards, she asked Brett to clarify his actions and gave encouragement like "you're on the right track."

In this first iteration of the PL debugging activity, the concurrent need to develop participants' roles (teacher/student and driver/navigator) had a determining impact on how the three teachers approached debugging. They positioned debugging as a distributed process where the driver has access to the DaSH and all possible inscriptions, the navigator offers next steps and resources, and the teacher (who knows what the issues are) clarifies and encourages. Brett's regular narration of his actions (e.g., "I can look through the code) and his thinking (e.g., "there's nothing on P10… so there's a problem") developed a representation of the process for his collaborators. Yet, Grace and David's access to important inscriptions were limited.

Episode B: Iteration 2 (Chris and Aaron)

In the second iteration of the debugging activity, Chris and Aaron worked without specific roles assigned and were given a written description of what the DaSH should do (*Expectation* below). In under ten minutes, they successfully got the DaSH's output to match the written Expectation. They initially took up distinct approaches to their collaboration: Chris narrated what he saw in the programming as compared to their Expectation and Aaron moved close to the DaSH and examined the physical setup. Collaboratively examining both parts of the system, Chris and Aaron determined the mismatch between the program's Pin and the Pin the sensor was connected to (Figure 3). They changed the program and downloaded the new program.

Figure 3
Transcript with screenshots of Chris and Aaron's production of a bug.



The excerpt in Figure 3 exemplifies the complex collaboration between Chris and Aaron. With Chris at the computer and Aaron at the physical setup, they layered inscriptions of the program, the Expectation, and the physical DaSH's wiring, finding the bug through joint work. For example, Chris' turn towards the setup in line 13 joined Aaron's location of the Pin (gesture during line 11) before he suggested they change the program to match the wiring (line 15). Their talk and gestures developed the association between inscriptions of the program and the physical setup to identify the issue. Generally, Chris and Aaron approached debugging as a collaborative, embodied process of collective noticing and decision making. While their emergent roles were somewhat static, they both had access to different inscriptions that, together, made clear what the issue was and what action to take to fix it. This experience has the possibility of developing pedagogies for both Aaron and Chris that support students building the embodied, and collaborative, skills of debugging such a physical computing system.

Discussion

Grace's question in Episode A ("do I need to know anything to be the teacher?") and her gained confidence once later knowing what the planted bugs were ("you're on the right track") highlights a key tension in learning to support debugging: teachers cannot know every issue that will arise and therefore need a different kind of confidence in supporting their students. DeLiema et al.'s (2019) teaching moves for supporting students each imply needing to know inquiry strategies, rather than the source of the issue. Offering teachers opportunities to work on authentic bugs created space for new teachers to develop inquiry strategies that they can model, prompt, and reflect on with their students. These strategies included different approaches to coordinating inscriptions of the program and wired system (physical system and wiring diagram) and changing the conditions of the system to test. Strategies emerged differently due to differences in PL activity structures. Neither iteration adequately offered teachers opportunities to consider the in-the-moment moves that supporting debugging requires.

In Episode A, the group was asked to model teachers and students doing pair programming. This created varied opportunities for Grace, David, and Brett to experience debugging. Changes in the PL activity for the second iteration led to participants engaging in a more collaborative way with the DaSH, as shown by Chris and Aaron's embodied joint work of uncovering the bug (Figure 3). Post activity reflections confirmed this change: participants' discussion in District 1 revolved around strategies and resources (e.g. programming tutorials, wiring diagrams) they might use to help their students debug. Due to time constraints, discussion in District 2 included participants sharing what bug they found and how. Reflection surveys showed participants found great value in the activity. Yet, they had little time to translate their experience to an evolving debugging pedagogy.

Teachers indicated in surveys and interviews that they want more experience working with the system to gain relevant knowledge and practices. Supporting teachers requires a balance of helping them feel experienced with the technology and prepared to support students' developing situated inquiry skills even when a student comes to them with an issue they have never seen. In both PL iterations, the activity and facilitation foregrounded the importance of finding the bugs, rather than explicitly understanding the strategies participating teachers employed. The first iteration offered some room for participants to consider teacher moves but much less ability to develop their own strategies. The second iteration offered ample opportunity for Chris and Aaron to develop strategies for debugging but offered little experience considering how to engage with students' strategies. Our proposed third iteration (Table 1) offers teachers debugging experience and time to consider pedagogies.

Conclusions and implications: Planning a third iteration

Teaching debugging necessitates finding comfort in not knowing the answer. Professional learning experiences about debugging need to balance supporting teachers in developing debugging experiences and skills with the pedagogical dimension of supporting students to become independent and collaborative debuggers. Both dimensions are important and professional learning experiences should be explicit about which they address and when. Our third iteration addresses the need to support teachers knowing *the strategies* for students to take in-the-moment, not the source of the problem. Teachers need experiences engaging with practicing debugging to develop their own "perception and action" (Flood et al., 2018) so they have relevant experiences with embodied collaboration, noticing inscriptions and making flexible decisions, which is structured in Round 1 (Iteration III). Teachers can then leverage their experiences as they model, rehearse, and reflect on debugging pedagogies that support students developing debugging strategies and mindsets (Round 2), rather than giving them "the answer."

References

DeLiema, D., Dahn, M., Flood, V. J., Asuncion, A., Abrahamson, D., Enyedy, N., & Steen, F. (2019). Debugging as a Context for Fostering Reflection on Critical Thinking and Emotion. In E. Manalo (Ed.), *Deeper Learning, Dialogic Learning, and Critical Thinking* (1st ed., pp. 209–228). Routledge.

Fields, D., Kafai, Y., Morales-Navarro, L., & Walker, J. (2021). Debugging by design: A constructionist approach to high school students' crafting and coding of electronic textiles as failure artefacts. *British Journal of Educational Technology 52*, 1078-1092.

Flood, V. J., DeLiema, D., Harrer, B. W., & Abrahamson, D. (2018). Enskilment in the digital age: The interactional work of learning to debug. In J. Kay & R. Luckin (Eds.), Proceedings of the 13th International Conference of the Learning Sciences (pp. 1405-1406). London: ISLS

Gendreau Chakarov, A., Biddy, Q., Hennessy Elliott, C., Recker, M. (2021). The Data Sensor Hub (DaSH): A Physical Computing System to Support Middle School Inquiry Science Instruction. Sensors.

Jordan, B., & Henderson, A. (1995). Interaction Analysis: Foundations and Practice. *Journal of the Learning Sciences*, 4(1), 39–103.

Latour, B. (1986). Visualization and cognition: Thinking with eyes and hands. Knowledge and Society: *Studies in the Sociology of Cultural Past and Present, 6*, 1–40.

Manz, E. (2015). Resistance and the Development of Scientific Practice: Designing the Mangle Into Science Instruction. *Cognition and Instruction*, 33(2), 89–124.

Acknowledgements

We are grateful for our SchoolWide Labs collaborators including Jennifer Jacobs, Quentin Biddy, Srinjita Bhaduri, Tammy Sumner, Bill Penuel, & Michael Schneider. This research was funded by the National Science Foundation (Award No. 1742053, No. 2019805 and No. 1742046) and the James S. McDonnell Foundation.