International Society of the Learning Sciences

Debugging Debugging Instruction: A Research-Practice Partnership in K-8 Computer Science Education

Andrea Wilson Vasquez, Code Savvy, awilsonvazquez@gmail.com

David DeLiema, Megan Goeke, Jeffrey K. Bye
ddeliema@umn.edu, goeke018@umn.edu, jbye@umn.edu
University of Minnesota

Abstract: Through a collaboration between a Minnesota-based CS education non-profit, learning scientists, and elementary and middle-school CS educators, we are collaborating to envision new ways to support students with the frequent impasses they experience when writing code. In this paper, we describe how we worked toward recognizing problems of practice, built on prior interaction analyses of classroom discourse, assembled a team of CS educators, developed (and critiqued) our shared language around debugging, articulated our rationale for particular learning designs, and calibrated our approach. We hope this paper serves as one concrete and generative example of how to use classroom-based interaction analyses as points of departure for empowering teacher-driven pedagogical imaginings.

Introduction

This paper provides a snapshot of an emerging research-practice partnership (RPP) between Code Savvy, a non-profit computer-science (CS) education organization (represented by Andrea Wilson Vasquez), Minnesota CS educators, and learning scientists studying debugging (David DeLiema, Jeff Bye, Megan Goeke). Our ongoing RPP centers CS educators in the discovery and specification of problems of practice and pedagogical approaches that draw inspiration from a previously underemphasized dynamic in debugging education: students and teachers select, modify, and discuss debugging pathways among many valid possibilities, including when noticing problems, positioning causes, and pursuing fixes (DeLiema et al., 2021).

Practitioner context: Code savvy and experience with debugging pedagogy

Our RPP is rooted in the work of Code Savvy, who has recognized that CS education is severely under-supported in Minnesota. Across the U.S., there is increasing demand for and interest in CS skills; but in Minnesota, where most of the people born in-state stay for their adult lives, there is a particular need to increase support for local development of a CS-informed community. Currently, Minnesota has no state standards regarding CS, and only 21% of Minnesota high schools offer any CS courses, placing Minnesota last of all 50 states in CS offerings (Code.org et al., 2022). Where CS is offered, these teachers tend to be the only CS educators in their school, district, or in some cases, geographic region. Unlike more common subjects, CS teachers in Minnesota cannot turn to their immediate teacher network for CS education support. Code Savvy was established in 2013 to address the specific, intersecting concerns in Minnesota CS education: lack of curricular guidance, isolation of CS teachers, and lack of access to CS educational opportunities, which collectively increase risk of inequitable CS experiences for Minnesota's young people. Driven by the goal of expanding equitable and engaging CS education, Code Savvy supports a network of CS educators across Minnesota ranging from full-time high school CS teachers to elementary teachers integrating CS into existing curriculum through a variety of professional development (PD) opportunities. The PD community serves as a support system for Minnesota's CS educators, acting as a place to articulate why to teach CS, how to teach CS, and how to navigate administrative structures. Currently 625 teachers strong, Code Savvy - with sustained leadership from Andrea between 2015 and 2023 - is uniquely positioned to both understand the needs and priorities of Minnesota's CS educators and to support dissemination of critical CS education findings.

Andrea's personal experience across 10 years invested in CS education is that debugging receives the least attention in PD sessions with CS educators. Debugging, which roughly speaking is the process of noticing problems, searching for causes, and implementing fixes, is an essential and common part of programming teaching and learning (McCauley et al., 2010), and a practice that is often backgrounded in formal CS education spaces (Perscheid et al., 2017). When she started teaching at a high school makerspace, Andrea had to develop techniques for teaching students how to learn from debugging, and recognized that supporting students to develop strong debugging skills was an essential part of the classroom's culture of belonging – as all students will experience moments of impasse during coding. Knowing that in Minnesota we have limited chances to engage students with CS, it is our ethical responsibility to ensure that when students are in our CS classrooms, they are fully supported

and included. Generating practical support for teaching debugging is directly related to expanding equitable and engaging CS education.

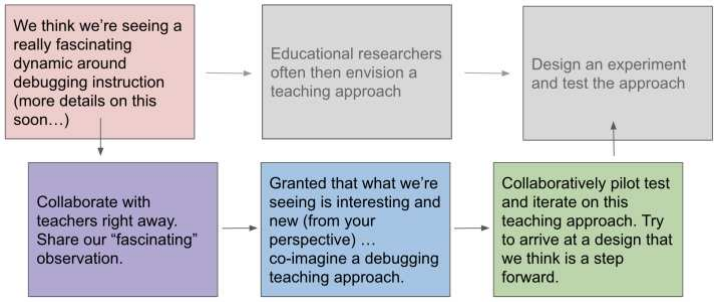
RPP origin story

Debugging was equally a topic of interest among our RPP's learning scientists. Attending to the existing debugging research literature (e.g., Fields et al., 2021) and using interaction analyses (e.g., Jordan & Henderson, 1995) of teacher-student discourse during debugging in naturalistics classroom settings, our earlier work had led to a framework that emphasized the open-ended process of teachers and students altering, foregrounding, and explicitly discussing different debugging pathways (DeLiema et al., 2021). That is, in many debugging situations, teachers and students have a wide range of pathways they can take through noticing problems, implicating causes, and pursuing fixes. For ease of reference, we refer to this as the debugging pathways framework. Andrea invited the learning sciences team to share the debugging pathways framework with the Code Savvy CS educator cohort she was leading, and we repeated this process annually over three years. Each time, we were struck by the CS educators' expansive reflections on debugging in ways that far-extended the framework and how it could inform pedagogy. Inspired by educational research that decenters research priorities and foregrounds participants' goals (e.g., Bang & Vossoughi, 2016), we pursued grant funding (see Acknowledgments section) and started to work together on a plan to turn these short sessions with Code Savvy cohorts into a sustained collaboration. We pursued this collaboration for several reasons: (a) debugging has persisted as a problem of practice both for teachers and educational researchers across several decades; (b) the pedagogical horizon opened up by the debugging pathways framework was vast and our team felt it was essential to center teachers' perspectives on where to take pedagogical designs; (c) our partner teachers were interested in investing time reflecting on debugging pedagogy; and (d) in our earlier Code Savvy sessions, teachers had both expressed interest in thinking with the debugging pathways framework and developed ideas about pedagogy that they signaled would be new to their classrooms and worth pursuing.

Moving into our RPP work, we held researchers' humility to educator expertise as a core value. Figure 1 is a screenshot of a flow chart we used to communicate this value and process to the CS educators in our RPP. We anchored this value in four additional design choices. First, in part to stave off too-early ideological convergence (e.g., Philip et al., 2017), we framed the debugging pathways framework as a draft fully open to revision. Second, we emphasized opportunities for educators to look at video data of programming classrooms and offer their own insights. While a rich tradition of video-viewing exists for teacher professional development (e.g., Sherin & van Es, 2005), we diverged from those efforts in that we did not have an intended learning goal or valued way of noticing classroom experiences in these data sessions. Third, we made concerted efforts to center teacher voice. All workshops prioritized time for our RPP leadership team to listen to the discussion and reflections of our CS educator collaborators. Fourth, we held back from quickly centering a specific problem of practice. That is, we viewed the debugging pathways framework as a provocation to both find new problems of practice and spark novel pedagogical approaches.

Figure 1

Framing in color our collaborative work in the RPP, contrasted with "top-down" research paths in gray.

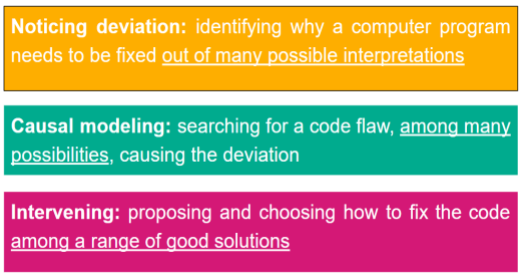


Workshop design

Our approach to workshop design reflects the tradition of participatory design-based research (Bang & Vossoughi, 2016), in which researchers develop pedagogical approaches in close collaboration with teachers, paying attention to problems of practice centered by educators, power dynamics within the team, and design solutions proposed by the educators, through iterative stages of implementation, data collection, and analysis. The workshops were designed as a year-long series with 5-7 CS teachers working with different aged students. Most of the teachers

who ended up joining had found out about the RPP because they had checked a box on a short survey expressing their interest in hearing about research opportunities following our earlier Code Savvy PD workshops. The remaining teachers knew Andrea through her CS education network. All teachers received stipends for their time in the workshops. In the first session, David and Jeff planned to introduce Figure 1 (above), make space to talk about the overarching goals of the RPP, and share the debugging pathways framework (see Figure 2 for a simplified version), which they had developed with Vijay Marupudi. This was meant to provide a perspective, language, and empirical focus on debugging that we hoped would unsettle the typical backgrounding of debugging in professional learning communities and provide a provocation for discussion about debugging pedagogy. In this way, the prior interaction analyses and framework (DeLiema et al., 2021) served as a point of departure for inquiry, not a pedagogical prescription. In sessions 2-4, we planned for our teacher collaborators to look at video data of students debugging, interpret the video, and discuss ideas for approaching scaffolding of student debugging. During these later sessions, we planned to have our RPP leadership team collaborate with the teachers to develop and iteratively revise conjecture maps: visual devices that provide an argument around a theory of design, short-term processes observable in the classroom, and valued long-term outcomes (Sandoval, 2014). This approach drew on prior work involving collaborative conjecture mapping in CS education (e.g., Lee et al., 2022). We planned to invite the teachers to pilot approaches in their own classrooms, documenting the results and their own reflections in iterations to the conjecture maps. After piloting, the entire RPP would have two additional sessions to consolidate the classroom pilots into a single unified conjecture map.

Figure 2
Debugging Pathways Framework



Implementation

Alongside 4 middle school CS educators and 1 first-grade CS educator, we have completed the first 4 workshop sessions and will soon begin classroom pilot work. In this reflection, we focus on two shifts in our process that took place during implementation and that are critical to the work of our RPP. First, we had intended to share the debugging pathways framework as a draft and then shift to envisioning pedagogical approaches. However, as the CS educators in our RPP reflected on the framework and applied this lens in video data sessions, the conversation kept returning to the framework as educators argued about its meaning and potential. In these arguments, CS educators critiqued both the focal terms in the framework (e.g., "my students would look at me sideways" if we used the word "deviation" in the classroom) and assumptions about the linear flow between these steps, even while embracing the notion that debugging in the framework was considerably more open-ended than traditional debugging models conveyed. To welcome this critique, David noted at the beginning of each workshop that we had put the "fascinating" interaction analysis observations from research in scare quotes in order to understand what the educators in our RPP made of these open-ended debugging dynamics. This shift pointed to fundamental questions about how to position interaction analysis in RPPs to stoke conversation while ensuring they are still treated as flexible representations in need of fine-tuning.

Second, we changed the structure of the fourth workshop from a group conversation to individual conversations with each CS educator. We had noticed that the framework sparked a wide range of learning design proposals, but the collaborative workshops did not provide enough space for each educator to unpack these imaginings; instead, discussions during group conversations leaned into contrasts between specific learning design proposals and contexts (e.g., noticing that an exercise wouldn't work in a particular classroom). In short, the substantial heterogeneity in learning design proposals left little space for deeper commonalities to be noticed and named in the moment. Shifting both our process and our goal, we moved toward individual conversations with CS educators to allow for each educator to articulate their thoughts in greater depth, without needing to connect to others' ideas in the moment. We introduced the conjecture map representation in each of these individual sessions and gave time to each educator to start assembling a process-based argument for why a particular debugging pedagogy might work and toward what ends.

Current results

We held a collaborative data session among the RPP leadership team following these individual meetings, with each leader 'representing' the conjecture map from one educator. We examined commonalities across disparate maps to glean a common pedagogical vision. Instead of arriving at a shared pedagogical approach (e.g., specific tools, curricula, pedagogical moves), we recognized shared lenses guiding specific designs. Megan further refined these notes into 4 lenses that stretched across all educators' unique pedagogical visions around debugging. (1) *Nonlinearity* captured how debugging involves engagement with – but not necessarily sequential progression between – the 3 framework steps. (2) *Future orientation* focused on making space to connect a current debugging solution to potential future codes and bugs. (3) *Community of support* addressed classroom teachers and students as a debugging support team. (4) *Multiplicity* focused on celebrating multiple, heterogeneous approaches to debugging. Our next step is to discuss these themes as a full RPP team, allow each educator space to assess whether their approach is captured by all 4 lenses (and what may be missing), and then consider what purchase they provide for refining each educator's unique pedagogy before classroom pilots.

Conclusion

Our RPP has embraced a process of flexibility and balance. Our approach has taken seriously a problem of practice noticed by a CS education non-profit and responded to a re-framing of debugging from learning sciences scholarship. At the same time, we made space to critique that prior research (including our own) and allow for considerably more individual expression of debugging pedagogies, even while we have worked to glean common lenses across these unique conjectures. As we work together to envision expansive forms of supporting debugging, we are committed to continuing to level the power between our non-profit leaders, learning scientists, and CS educators, working toward a common, ambitious vision for supporting students that makes possible individual forms of pedagogical expression and fully embraces the expertise of educators.

References

Bang, M., & Vossoughi, S. (2016). Participatory design research and educational justice: Studying learning and relations within social change making. *Cognition and Instruction, 34*(3), 173-193.

Code.org, CSTA, & ECEP Alliance (2022). *2022 State of Computer Science Education: Understanding Our National Imperative.* Retrieved from https://advocacy.code.org/stateofcs

DeLiema, D., Bye, J. K., & Marupudi, V. (2021). *Programming instructors and students' active (and partial) debugging: Deviation noticing, causal modeling, and intervening*. Paper presented at the American Educational Research Association, Virtual Meeting, Virtual Annual Meeting.

Fields, D. A., Kafai, Y. B., Morales-Navarro, L., & Walker, J. T. (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*(3), 1078-1092.

Jordan, B., & Henderson, A. (1995). Interaction analysis: Foundations and practice. *The Journal of the Learning Sciences, 4*(1), 39-103.

Lee, U.-S., DeLiema, D., & Gomez, K. (2022). Equity conjectures: A methodological tool for centering social change in learning and design. *Cognition & Instruction, 40*(1), 77-99.

McCauley, R., Fitzgerald, S., Lewandowski, G., Murphy, L., Simon, B., Thomas, L., & Zander, C. (2008). Debugging: A review of the literature from an educational perspective. *Computer Science Education, 18*(2), 67–92.

Perscheid, M., Siegmund, B., Taeumel, M., & Hirschfeld, R. (2017). Studying the advancement in debugging practice of professional software developers. *Software Quality Journal, 25*(1), 83-110.

Philip, T. M., Gupta, A., Elby, A., & Turpen, C. (2018). Why ideology matters for learning: A case of ideological convergence in an engineering ethics classroom discussion on drone warfare. *Journal of the Learning Sciences, 27*(2), 183-223.

Sandoval, W. (2014). Conjecture mapping: An approach to systematic educational design research. *Journal of the Learning Sciences, 23*(1), 18-36.

Sherin, M., & van Es, E. (2005). Using video to support teachers' ability to notice classroom interactions. *Journal of technology and Teacher Education, 13*(3), 475-491.

Acknowledgments

We express our deep thanks and our excitement to continue our collaboration with the CS educators in our research-practice partnership. We also thank the UMN Grant-in-Aid program for providing funding to support this collaborative work.