



Computational Thinking: A Tale of Debugging

Ali D. Asif, University of Massachusetts Dartmouth, aasif@umassd.edu
Hamza Malik, University of Massachusetts Dartmouth, hmalik1@umassd.edu
Chandra H. Orrill, Rethink Learning Labs, chandra.orrill@rethinklearning.com
Stephen B. Witzig, University of Massachusetts Dartmouth, switzig@umassd.edu
Ramprasad Balasubramanian, University of Massachusetts Dartmouth, r.bala@umassd.edu
Shakhnoza Kayumova, University of Massachusetts Dartmouth, skayumova@umassd.edu

Abstract: This exploratory research analyzes the video-recorded data of four elementary-grade teachers debugging a school tour activity while utilizing a programmable robot, Photon. This summer's professional development session on computational thinking (CT) integration was four hours long and was focused on debugging as a key CT component. The results indicate that teachers worked collaboratively to debug their way through errors using different strategies, such as step-by-step execution or incremental code development.

Introduction

A bug is a noun that means a defect or fault in a machine, plan, or the like, and to debug is to eliminate such defect (Shapiro, 1987). Debugging is a critical component of CT, providing a systematic approach to dealing with bugs or errors (Rich et al., 2019). Evidence from research suggests that debugging programs or codes can enhance general troubleshooting skills in all other non-programming domains (Michaeli & Romeike, 2019). This signifies an immediate need to help elementary grade teachers and support them in bringing debugging into their classrooms (Haduong & Brennan, 2019). This study explores how four elementary-grade teachers use debugging in a social and collaborative environment.

Conceptual framework and methodology

For this exploratory study (Maxwell, 2013), our guiding research question was, "In what ways, if any, do two groups of elementary grade teachers practice debugging while programming a robot to automate a school tour activity?". Borrowing from the larger framework of Community of Practice by Wenger (1998), we have used three dimensions of practice (Asif et al., 2023) to analyze the collaborative work of participants within three dimensions of debugging (Rich et al., 2019). The interlocking dimensions of practice include mutual engagement, joint enterprise, and shared repertoire (see Wenger, 1998). The three dimensions of debugging include strategies to find and fix bugs, types of bugs encountered, and the role of bugs (see Rich et al., 2019). Our data include four hours of video recording of a face-to-face professional development (PD) facilitated by our PD partner, Eduscape (<https://www.eduscape.com>). We adapted two coding phases from Charmaz's (2014). The initial phase included generating descriptive codes, and the focused phase included connecting themes with our conceptual framework.

Findings and discussions

As part of the PD, teachers were asked to work in groups to write step-by-step instructions or pseudocode to design an optimal school tour activity utilizing a programmable robot, Photon (<https://photon.education>). Initially, teachers encountered bugs in their coding related to wrong directions and incomplete instructions. They identified these bugs through mutual accountability of negotiating about the bugs, i.e., rereading the activity instructions together. Their strategy to find and fix the bugs was a step-by-step execution of instructions, where one participant read the code, and the other acted on it, mimicking the movement of Photon. In the next phase, teachers incrementally developed their code using step-by-step instructions execution and using the intermediate results as a checking process. Lastly, teachers also found bugs embedded in the code's implementation and logic of the programming models. They explained the bug's location, its behavior, and possible ways to debug it in their own shared repertoire, i.e., reporting when their code checked for a particular condition only at the start. This is an example of debugging as a social act of collaborative thinking and participation in a shared activity (Cadwell et al., 2022). Our results highlight relationship patterns between types of bugs and the strategies employed by the teachers to find and fix them (Liu et al., 2017). Teachers reproduced previous results to understand the bug and the actual working of the robot. They tested isolated pieces of their code for validation in combination with step-by-step execution.

Conclusion



Debugging as a field of research has been overlooked (Lieberman, 1997), and it is an underrepresented topic in K-12 classrooms (Rich et al., 2019). Teachers of K-12, especially elementary grades, often do not know debugging as a systematic process, and explicit connections with debugging strategies are usually missing in their lessons (Michaeli & Romeike, 2019). This research suggests that teachers' strategies for finding and fixing bugs varied as the bugs became more challenging (Liu et al., 2017). We also found that as teachers worked together on the problem, they started using multiple strategies to find and fix bugs simultaneously, reflecting on the interplay (Rich et al., 2019) between different strategies for finding and fixing bugs.

Implications

Historically, CT has been framed using cognitive framing strategies (Kafai & Proctor, 2022), which is also true in research focused on debugging (Micaeli & Romeike, 2019). Therefore, one of the significant implications of this research study is that it uses the lens of practice, grounded in the socio-cultural theory of Community of Practice (Wenger, 1998), to interpret the debugging process in a collaborative and shared environment. Another conceptual implication is viewing the systematic debugging process via three dimensions (Rich et al., 2019): reporting on the types of bugs encountered by teachers in block-based programming, the strategies used to find and fix them, and the role of bugs in improving design. This research explores the collaborative debugging process of elementary grade teachers in the hope that the findings of this research study will inform other researchers of similar interests.

References

- Asif, A. D., Malik, H., Orrill, C. H., Balasubramanian, R., & Kayumova, S. (2023, April 18 - 21). *An Exploratory Study: Understanding Teachers' Use of Decomposition*. Annual Meeting National Association for Research in Science Teaching, Chicago, IL, United States.
- Cadwell, P., Federici, F., & O'Brien, S. (2022). Communities of practice and translation: An introduction. *JoSTrans: the Journal of Specialised Translation*, (37).
- Charmaz, K. (2014). *Constructing grounded theory*. Sage.
- Haduong, P., & Brennan, K. (2019, February). Helping K-12 teachers get unstuck with Scratch: the design of an online professional learning experience. In *Proceedings of the 50th ACM technical symposium on computer science education* (pp. 1095–1101). Minneapolis, MN: ACM.
- Kafai, Y., Hutchins, N., Snyder, C., Brennan, K., Haduong, P., DesPortes, K., DeLiema, D., Aalst, O. W., Flood, V., Fong, M., Fields, D., Gresalfi, M., Brady, C., Steinberg, S., Franklin, D., Etinger, D., Coenraad, M., Palmer, J., Weintrop, D., ... Bulalacao, N. (2020). Turning Bugs into Learning Opportunities: Understanding Debugging Processes, Perspectives, and Pedagogies. In Gresalfi, M. and Horn, I. S. (Eds.), *The Interdisciplinarity of the Learning Sciences*, 14th International Conference of the Learning Sciences (ICLS) 2020, Volume 1 (pp. 374–381). Nashville, Tennessee: International Society of the Learning Sciences.
- Kafai, Y. B., & Proctor, C. (2022). A revaluation of computational thinking in K–12 education: Moving toward computational literacies. *Educational Researcher*, 51(2), 146–151.
- Lieberman, H. (1997). The debugging scandal and what to do about it. *Communications of the ACM*, 40(4), 26–30.
- Liu, Z., Zhi, R., Hicks, A., & Barnes, T. (2017). Understanding problem solving behavior of 6–8 graders in a debugging game. *Computer Science Education*, 27(1), 1–29.
- Maxwell, J. A. (2013). *Qualitative research design: An interactive approach*. Sage publications.
- Michaeli, T., & Romeike, R. (2019). Improving debugging skills in the classroom: The effects of teaching a systematic debugging process. *Proceedings of the 14th Workshop in Primary and Secondary Computing Education* (pp. 15:1–15:7). ACM. <https://doi.org/10.1145/3361721.3361724>
- Rich, K. M., Strickland, C., Binkowski, T. A., & Franklin, D. (2019, February). A K-8 debugging learning trajectory derived from research literature. *SIGCSE 19 - Proceedings of the 50th ACM Technical Symposium on Computer Science Education*, 745–751. <https://doi.org/10.1145/3287324.3287396>
- Shapiro, F. R. (1987). Etymology of the computer bug: History and folklore. *American Speech*, 62(4), 376–378.
- Wenger, E. (1998). *Communities of practice: Learning, meaning, and identity*. Cambridge University Press.

Acknowledgment

This material reported here is based upon work supported by the National Science Foundation (NSF) under Grant 2331742. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the NSF.