International Society of the Learning Sciences

Computational Thinking: Teachers' Practice of Abstraction

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@rethinkinglearning.com
Ramprasad Balasubramanian, University of Massachusetts Dartmouth, r.bala@umassd.edu
Shakhnoza Kayumova, University of Massachusetts Dartmouth, skayumova@umassd.edu

Abstract: Elementary grade teachers are often not fully prepared to teach a computing-rich curriculum, and the demand of the digital age to integrate Computational Thinking (CT) into their classrooms has put them at a challenge. Under the larger umbrella, abstraction lies at the heart of CT. Abstraction allows moving between various information levels while targeting complex problems and creating rich design solutions. This study focuses on how one pair of elementary-grade teachers collaborated, using abstraction to solve a maze challenge, helping each other move between different layers of information. The videotaped data of one day of teachers' professional development was analyzed through three dimensions of Community of Practice (CoP). Results suggest that through mutual engagement in pursuing a joint enterprise and their shared repertoire, elementary-grade teachers moved their focus between different levels of abstraction simultaneously and effectively.

Introduction

There is a growing demand to integrate computational thinking (CT) into all disciplines, starting from elementary grades, especially in mathematics and science (Dong et al., 2019; Kafai & Proctor, 2022). This increasing emphasis on including CT practices has challenged elementary-grade teachers. Most of them have limited Computer Science (CS) knowledge, and concepts associated with CT are difficult for them to understand (Yadav et al., 2017). Teachers lack the knowledge and confidence to create learning activities to support students in developing CT (Kyumova et al., 2023). Under the larger umbrella of CT practices, such as algorithmic thinking, decomposition, and debugging (see Grover & Pea, 2013; Weintrop et al., 2016), researchers consider abstraction as a fundamental idea of CS (Statter &Armoni, 2020) and center it at the heart of CT (Wing, 2008; Rijke et al., 2018). To capture the true essence of abstraction in CT, it is essential to understand how computer scientists employ this concept to solve problems and design complex systems (Wing, 2006). In CS, abstraction involves moving between various levels of information simultaneously while focusing on what is relevant to that level. Like using a zoom lens to change the resolution and concentrate only on what is essential to solve the part of the problem at hand (Colburn & Shute, 2007) and then zooming out to observe other related parts (Asif et al., 2024) of the problem. Past research in education sheds little light on how elementary-grade teachers use abstraction while solving problems. Even less is known about how abstraction can be viewed and analyzed as collaborative work in a social setting. Heeding the call of Kafai and Proctor (2021) to broaden the boundaries of research and to incorporate different framings of CT, this research conceptualizes teachers' use of abstraction as collaborative work by using the lens of three dimensions of practice in a community of practice (Wenger, 1998). The guiding question of this research is: How does one group of elementary-grade teachers use abstraction to solve a maze challenge? We analyzed video-recorded data of two teachers working together on the maze challenge. The results suggest that through the mutual engagement of teachers in a joint enterprise, they frequently moved between multiple layers of information using abstraction as a zoom lens to toggle focus between the problem as a whole and different parts of the problem (Statter &Armoni, 2020). We assert that by exposing teachers to tasks that can provide them an opportunity to use abstraction, we can help them grasp the difficult-to-understand idea of simultaneously moving between different layers of information (Statter & Armoni, 2020). Additionally, it can help them create similar learning opportunities in classrooms for their students, making them better situated to integrate CT in their lessons.

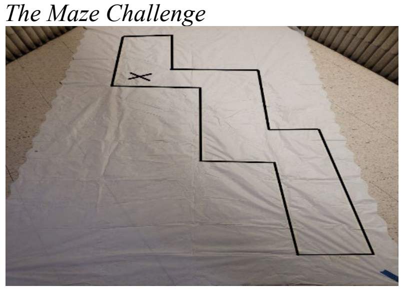
Conceptual framework and review of relevant literature

The abstraction process in CS produces layers of information, and we work with multiple layers simultaneously (Wing, 2008). This layered information structure helps build sub-models of a more extensive system that can be developed and evolved independently (Colburn & Shute, 2008). Abstraction, viewed from this perspective, takes the form of a zoom lens (Statter & Armani, 2020), zooming in to filter out irrelevant information while focusing on one part of the system and zooming out to include all related information and seeing a system working as a whole. This study frames abstraction from the lens of CoP, a social learning theory that considers learning as a social participation (Wenger, 1998). Within the larger framework of CoP, the practice has three interlocking dimensions. Mutual engagement as when people share information and ideas and build mutual relations by sharing ideas and information. Joint enterprise as shared relations of mutual accountability while pursuing a common goal and a shared repertoire, include words, tools, actions, and ways of doing things. We have used three dimensions of practice to interpret the collaborative use of abstraction by the teachers.

Methodology

This study utilized an intrinsic case study (Stake, 1995) of two elementary-grade teachers using abstraction while working together on a maze problem. The guiding question for this research was: How does one group of elementary-grade teachers use abstraction to solve a maze challenge? Data constitute one day of a year-long, face-to-face Professional Development (PD) of elementary grade teachers to help them integrate CT into their classroom lessons. The 90-minute session was video recorded. Our PD partner, Eduscape (https://www.eduscape.com), facilitated the PD session. The research participants were two elementary-grade teachers, Lauren and Molly (pseudonyms). Molly was a third-grade teacher, and Lauren was Science Technology Engineering Mathematics (STEM) coach in the same school in the Northeast of the United States (US). The teachers were tasked with programming a robot, Photon (https://photon.education), to navigate a maze challenge (see Figure 1). Both participants worked in groups to solve the challenge using the programmable robot. We used deductive coding (Lauckner et al., 2012) to analyze the data. The analysis was direct-interpretive (Stake, 1995) using predefined codes from the conceptual framework: mutual engagement, joint enterprise, and shared repertoire. We also coded instances of abstraction when participants changed their focus from one part of the problem to another and when they zoomed out their focus to look at the larger picture of the challenge to test their solutions.

Figure 1
The Maze Challenge



The PD facilitator introduced the maze challenge. In addition to navigating through the maze, the challenge also included coding the robot to utter a sound and/or change color when it reached a point marked as X on the maze. Once finished with the mark X, the robot could proceed to the end of the maze. The teachers used the block-based coding option in the programming environment for this activity.

Findings and discussions

Findings indicated that both participants of this research study used multiple levels of abstraction (Statter & Armani, 2020) as they zoomed into different parts of the problem, such as navigating the robot through the maze accurately and doing something exciting at the X marker. From the outset, Molly took the responsibility for measuring the distance between the turns and placing the robot at the start point after each try. Lauren took charge of building the code on the iPad. To solve the maze, they mimicked the moves the robot needed to make by walking beside it. They estimated the distances, noted the direction of each turn, then started coding the robot. After the first attempt, both started working on two different levels of information: Lauren with accurate movements to navigate the robot through the maze, while Molly on the idea to make the robot do something on the “X”. Once they completed the maze challenge, they continued to improve the code by bringing accuracy to the robot's movements and adding more details to the robot’s celebrations. Their engagement brought forth shared ideas and tools, mutually negotiated meanings, and self-accountability (Wenger, 1998), which paved the way for them to move freely between multiple layers of abstraction.

<u>Mutual engagement</u>

Molly was interested in “doing something on X”. Lauren walked her through different options that they could do as a celebration upon getting to the X. They then moved the focus from the bigger picture of the problem, where they not only had to maneuver the robot through the maze but also make a noise or color change at different markers and concentrated only on the information relevant to maneuvering the robot with accuracy. This indicated they could zoom into one part of a complex problem, and everything associated with that part only. Once the maneuvering was correct, they shifted their focus to "doing something on X" and turned it into a celebration dance. When the code was tested, they felt the need to improve it. They had to adjust their measurements of distance and celebrations multiple times before agreeing to complete the challenge successfully. They moved their focus multiple times between these parts of the problem and the testing phase. In the end, they wanted to make more improvements to their celebrations. Even after announcing victory, Molly was curious to make the robot dance faster by changing the movement speed, but she could not complete it due to time constraints. This change of focus between different parts of the challenge is a key example of abstraction as a zoom lens (Statter & Armani, 2020). Participants' mutual engagement in solving the task led them to take charge of their own negotiated roles, helping each other to move between different levels of abstraction.

Joint enterprise

Both teachers wanted to improve and expand their code after each attempt. Their first execution of the solution was complete. The robot stayed almost within the boundaries and celebrated on point X. But Lauren and Molly were not satisfied, wanting more accuracy and adding more to the robot's celebration. This sense of mutual accountability (Wenger, 1998) pushed them to see a single problem having different layers of information (Statter & Armani, 2020; Wing, 2008), one related to maneuvering the robot through the maze and the second to the celebration dance. With each attempt, they could move between these layers more freely—thus demonstrating a key example of abstraction.

Shared repertoire

In the beginning, when they faced a problem with approximating distance, Lauren proposed using a ruler. Later, Molly found a way to measure distance with the robot by breaking down the code into parts by incrementally adding the code after testing the previous. Molly was particularly interested in the celebration dance of the robot, and Lauren shared the possibilities that could be programmed for that. The teachers brought their tools to measure the distance, used words with their own negotiated meaning, and took actions that helped them see a single complex problem as a layered and structured design. The participants' shared repertoire (Wenger, 1998) helped them use abstraction more frequently, zooming in and out to focus only on relevant information for the part in hand.

Conclusion

The result suggests that these teachers utilized CS abstractions regularly when provided with the opportunity. They switched the resolution of their focus on the task at hand depending on the need and worked between multiple layers of information (Colburn & Shute, 2007; Wing, 2008). The results also signify the three interlocking dimensions of practice (Wenger, 1998) in the participant's work. Their mutual engagement and accountability, as well as their tools and actions, helped them to work simultaneously at multiple levels of abstraction. Wing (2008) argued that abstraction in computer science is different, richer, and more complex than that of mathematics and physical science. According to her, the abstraction process in computer science produces layers, and we work with two or more layers simultaneously. We also found similar results as the participants broke down the challenge into manageable parts (Asif et al., 2023). Both took on different responsibilities and layered their execution steps into one code, tested it, and then went back to improving their code, making it more accurate and turning the celebration of the robot into a more complex dance.

Implications

An increased requirement of the digital age to bring CT to young learners (Qualls & Larry, 2010; Wing, 2006; Yadav et al., 2014) has put math and science teachers at a challenge (Dong et al., 2019; Weintrop et al., 2016). It is even more challenging to create learning opportunities for elementary-grade students to practice using abstractions and make this tool a regular part of their vocabulary. This research has practical implications as such activities can be adapted in classrooms. CT is mainly viewed as a set of mental tools, a cognitive ability, and an individual activity. Kafai and Proctor (2021) highlighted the need for time to broaden the theoretical boundaries (Kafai & Proctor, 2021). This study is one of such attempts where abstraction, a key component of CT (Dong et al., 2019; Weintrop et al., 2016; Wing, 2008), is analyzed as a collaborative activity through the lens of three dimensions of practice within a social learning setting (Wenger, 1998). This research calls for viewing CT practices (Weintrop et al., 2016) as a work of collaboration instead of individual activity. This research also holds conceptual implications for abstraction. Computer scientists use abstraction to focus on different layers of information (Colburn & Shute, 2008) related to different parts of the problem, which are related to each other (Asif et al., 2023). Little research has been done explicitly exploring abstraction in elementary settings. This work begins to address that literature gap. While the field agrees that it is essential to conceptualize abstraction as a recursive process to solve problems that have structural similarities but their surface details match (Weintrop et al., 2016), this study addresses to explicate what these processes look like outside of CS and for a broader range of audiences, including elementary teachers. We also assert that providing teachers an opportunity to use CS abstraction will also help them create similar learning opportunities for their students.

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.

Asif, A. D., Malik, H., Stronach, R., Witzig, S. B. & Balasubramanian, R., (2024, March 6). *Technology Box: A STEM-Integrated Modeling Activity for Students, Teachers, and Environmental Educators.* Annual Conference Massachusetts Environmental Education Society, Worcester, MA, United States.

Colburn, T., & Shute, G. (2007, June 5). Abstraction in computer science. *Minds and Machines, 17*(2), 169-184. https://doi.org/10.1007/s11023-007-9061-7

Dong, Y., Catete, V., Jocius, R., Lytle, N., Barnes, T., Albert, J., Joshi, D., Robinson, R. & Andrews, A. (2019, February). PRADA: A practical model for integrating computational thinking in K-12 education. *Proceedings of the 50th ACM Technical Symposium on Computer Science Education - SIGCSE '19,* 906–912. https://doi.org/10.1145/3287324.3287431

Grover, S., & Pea, R. (2013, January 1). Computational thinking in K–12: a review of the state of the field. *Educational Researcher, 42*(1), 38-43. https://doi.org/10.3102/0013189X12463051

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.

Kayumova, S., Asif, A. D., Richard, E., Orrill, C. H., Liu, Z., Gearty, Z., Thapa, R., Tasnim, N., Balasubramanian, R. (2023 April 13-16). *Exploring elementary teachers' eagerness and reluctance to integrating computational thinking*. Annual Meeting American Educational Research Association, Chicago, IL, United States.

Lauckner, H., Paterson, M., & Krupa, T. (2012, March). Using constructivist case study methodology to understand community development processes: proposed methodological questions to guide the research process. *The Qualitative Report, 17*(13), 1-22. https://doi.org/10.46743/2160-3715/2012.1790

Qualls, J. A., & Sherrell, Larry. B. (2010, May). Why computational thinking should be integrated into the curriculum. *Journal of Computing Sciences in Colleges, 25*(5), 66-71.

Rijke, W. J., Bollen, L., Eysink, T. H., & Tolboom, J. L. (2018). Computational thinking in primary school: An examination of abstraction and decomposition in different age groups. *Informatics in Education, 17*(1), 77-92. https://doi.org/10.15388/infedu.2018.05

Stake, R. E. (1995). *The art of case study research*. Sage.

Statter, D., & Armoni, M. (2020, March). Teaching abstraction in computer science to 7th grade students. *TOCE -ACM Transactions on Computing Education, 20*(1), 1-37. https://doi.org/10.1145/3372143

Weintrop, D., & Wilensky, U. (2017). Comparing block-based and text-based programming in high school computer science classrooms. *ACM Transactions on Computing Education (TOCE), 18(*1), 1-25.

Wenger, E. (1998*). Communities of practice: Learning, meaning, and identity*. Cambridge University Press.

Wing, J. M., (2006). Computational thinking. *Communications of the ACM 49*(3), 33–35

Wing, J. Marry., (2008, July 31). Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 366*(1881), 3717-3725. https://doi.org/10.1098/rsta.2008.0118

Yadav, A., Good, J., Voogt, J., & Fisser, P. (2017). Computational thinking as an emerging competence domain. In M. Mulder (ed.), *Competence-based vocational and professional education, Technical and Vocational Education and Training: Issues, Concerns and Prospects, 23*, 1051-1067. Springer, Cham. https://doi.org/10.1007/978-3-319-41713-4_49

Acknowledgments

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.