Making Abstraction Concrete in the Elementary Classroom

Eping E. Hung
Computer Science
Southern Oregon University
Ashland, OR, USA
hunge@sou.edu

Maggie Vanderberg
Computer Science
Southern Oregon University
Ashland, OR, USA
vanderbem@sou.edu

Gladys Krause
School of Education
William & Mary
Williamsburg, VA, USA
ghkrause@wm.edu

Eva Skuratowicz
Division of Social Sciences
Southern Oregon University
Ashland, OR, USA
skuratoe@sou.edu

ABSTRACT

In recent years, several research projects have introduced elementary school teachers to computational thinking as a first step in familiarizing students with computer science concepts at an early age. A consistent challenge reported in these initiatives is teaching abstraction. This position paper offers preliminary recommendations for abstraction pedagogy in elementary education. These suggestions stem from an analysis of unplugged abstraction examples showcased during a summer institute on computational thinking.

By examining commonalities among abstraction examples, key parts of the process of abstraction pertinent to elementary classrooms were identified: (1) the abstraction process is typically performed in reverse since students in elementary school are given abstractions to start with; (2) evaluation of concrete details to support an abstraction is part of the filtering step of abstraction; (3) in the absence of evaluation criteria, pattern recognition can be applied to a set of concrete examples to extract characteristics of an abstraction; and (4) abstractions can be supported by not only concrete details but other abstractions which students will need to develop an understanding of before fully comprehending the initial concept.

Preliminary recommendations for abstraction instruction include having students evaluate examples; engaging students in pattern recognition to extract characteristics of an abstraction; developing student fluency in describing abstractions, their supporting examples, and characteristics; and assessing students by asking not only for examples of abstractions, but for their characteristics as well.

SIGCSE 2024, March 20–23, 2024, Portland, OR, USA

ACM 979-8-4007-0423-9/24/03$15.00
https://doi.org/10.1145/3626252.3630890

CCS CONCEPTS

•Social and professional topics~Professional topics~Computing education~Computational thinking•Social and professional topics~Professional topics~Computing education~K-12 education

KEYWORDS

Professional development, Computational thinking, Unplugged computational thinking, Abstraction, Elementary education

ACM Reference format:

Eping E. Hung, Maggie Vanderberg, Gladys Krause, and Eva Skuratowicz. 2024. Making abstraction concrete in the elementary classroom. In *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1 (SIGCSE 2024), March 20--23, 2024, Portland, OR, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3626252.3630890

1 INTRODUCTION

Wing's call to teach computational thinking (CT) in schools [1] ignited a series of research projects introducing CT through teaching [2,3,4,5]. Through these projects teachers across the nation have engaged in professional development (PD) to learn how to teach CT in hopes of promoting the use of these skills in their students. One repeated theme across these PD projects is that elementary school teachers report difficulty teaching the CT concept of abstraction across the curriculum [2].

Our conjecture is that the difficulty teachers have in teaching abstraction across the curriculum stems from a gap in abstraction literature. There is a rich history of research describing abstraction [6], and a more recent focus on identifying pre-existing examples of abstraction in the elementary classroom in various core subjects [7]. But the research has yet to identify a clear abstraction process that educators can use across the core subjects of the elementary curriculum.

In this paper, we outline a series of abstract concept maps utilized to organize worked examples showcasing abstraction across various subjects within the elementary classroom. Through a comparative analysis of these abstract concept maps, we identify a unifying process of abstraction that spans the curriculum and distill the essence of abstraction for elementary classrooms. Our paper presents preliminary recommendations for a cohesive abstraction pedagogy tailored for elementary education.

2 ABSTRACTION

Abstraction is considered essential to computational thinking [8] and a key skill for success in computing [9]. While simple definitions of abstraction exist, continued scholarship regarding abstraction has revealed many complex details that at once invite and resist synthesis [6, 10].

In its most basic incarnation, abstraction is the process of creating a simplified representation from more complex detail; it is also the end-product of such a process, so simplified representations can themselves be called abstractions [11]. Examples of abstractions include words [12], which contain detailed meaning; pictures or drawings, which can be representations of three-dimensional real-world objects [13] (or of a thousand words, as the saying goes); and computer simulations, which can model large complex systems or phenomena [5, 14].

Abstractions exist on a continuum, and an abstraction at the correct level, where lower levels contain more detail and higher levels contain less detail, can communicate just the right information for an intended purpose [15]. Conversely, the process of abstraction gone awry can result in an abstraction containing too little or too much detail [16].

2.1 Filtering vs. Generative Processes

The process of abstraction often results in a category or generalization about a set of details [17] or captures the *essence* of a set of details [18]. These results can be achieved through a filtering or generative process.

Abstraction through filtering can be accomplished through two variations: through *decomposition* to isolate meaningful data and exclude less important details; and via *generalization*, which involves applying pattern recognition across a set of similar data and finding commonalities to create a category [9]. The combination of these main variations is called *extraction* and is similar to Piaget's idea of *empirical abstraction* [19], which is abstraction that derives knowledge exclusively from a person's observations [18]. In CT literature, abstraction is most commonly defined as the filtering of details [8, 18].

Less commonly emphasized in CT literature, but perhaps more powerful [18] is the idea that abstraction can be *generative:* the process of abstraction can create new meaning that was not previously understood, perceived, or even present in the original details [20]. This ties in nicely with Piaget's *reflective abstraction* where the process of reflecting on lower-level observations and experiences results in new higher-level understanding [19]. A good example of reflective abstraction can be found in the poem "The Red Wheelbarrow" by William Carlos Williams [1938/1986]:

so much depends
upon

a red wheel
barrow

glazed with rain
water

beside the white
chickens

Simply applying empirical abstraction to this poem would merely result in the understanding that the poem is about "wet wheelbarrows and chickens." However, applying reflective abstraction to the poem generates a wholly new idea which did not even exist as a detail to highlight: that reflective abstraction is crucial to transforming our observations into understanding of the greater themes of the world around us.

2.2 Other Abstraction Variants

More specific variations of abstraction exist and are often associated with a particular context. *Metaphor* is an abstraction suffused with meaning in literature, but also in computer science, affording readers and users greater understanding of the story or sub-system to which the metaphor is applied [22]. Abstraction as *chunking* is mentioned in educational theory [23] and in computer science [24] to allow students and programmers quick access to a larger set of knowledge or programmatic behavior through a simplified concept or defined function.

2.3 Abstraction in Computer Science

Abstraction in the context of computer science displays additional characteristics: the existence of *abstraction layers* [8] and the need for abstraction as *structure identification* [24]. Layers of abstraction are a type of abstraction that breaks a large complex system into a hierarchical structure, where each layer contains its own *semiotic register*—a particular vocabulary and specific rules [20]. Relationships between layers are further abstracted through structure identification, revealing the nuances of how layers or decomposed parts should interact with each other [24]. Through abstraction layers, computer scientists can separate their concerns and focus on a single layer of a complex system and move easily between layers if necessary.

Similar sounding to abstraction layers, yet drastically different, a hierarchy of *abstraction levels* defined by Perrenet, Groote, and Kaasenbrood [2005], commonly known as the PGK-hierarchy, describes four abstraction levels a programmer must freely and expertly move between to develop a successful program: the *problem*, *algorithm*, *code*, and *execution* level. While abstraction layers in computer science describe how multiple technologies or designs are built one on top of another and interact with each other, these abstraction levels describe different perspectives through which a single computing solution must be analyzed.

3 THE IRONY AND PARADOX OF ABSTRACTION

A listing of all the vocabulary associated with descriptions of abstraction in literature is enough to overwhelm even the most seasoned academic, let alone an elementary teacher learning about abstraction for the first time: *decomposition, generalization, empirical abstraction, generative, reflective abstraction, metaphor, chunking, abstraction layers, semiotic register, structure identification, and abstraction levels.* It is unclear which is the greater irony: that abstraction itself is in need of abstraction or that computer scientists—themselves experts in applying abstraction to develop elegant solutions for a multitude of complex computing problems—struggle to develop an elegant definition of abstraction. There may be a clear reason behind this struggle, which is the dependence of abstraction on context and purpose [15]. How a person uses and understands abstraction will depend entirely on whether a particular problem or context calls for extraction, generative abstraction, or creating abstraction layers for a large system. Thus, one act of abstraction may look completely different in one context than in another.

The paradox of creating a single unified definition of abstraction is that a complete definition of abstraction contains so many elements as to be abstruse and that abstraction would be needed to simplify the definition. Yet, a simplified and likely decontextualized definition of abstraction might be useless in practice, since application of abstraction would happen in a specific context, and thus would require context-specific details that had been abstracted out during the creation of the simplified definition of abstraction. It may be that abstraction needs a definition for each context in which it is applied.

4 ABSTRACTION IN ELEMENTARY EDUCATION

This position paper presents a set of recommendations for abstraction pedagogy in elementary education based on preliminary observations from an attempt to teach abstraction to teachers. To create a targeted learning experience, researchers asked teachers to first identify problems of practice that could be solved with abstraction. Those problems were then used as models in which to apply abstraction and derive exactly what aspects of abstraction were used in solving them. Lastly, pattern recognition was applied across the set of abstraction solutions to abstract a precise understanding of abstraction specific to the context of elementary education.

4.1 Context

The work presented here was developed in the midst of a five-day unplugged computational thinking professional development (PD) summer institute during the first year of a three-year CSforAll:Research Practice Partnership. Twenty elementary teachers from three school districts in the Pacific Northwest region of the United States were given a two-day introduction to CT and four core CT concepts: pattern recognition, algorithms, decomposition, and abstraction, also known as the PRADA elements [2]. In the remaining three days, teachers were tasked with creating lessons for the ensuing school year that integrated one or more of the four CT concepts into the four core subjects of English language arts, mathematics, social studies, and science, as well as social-emotional learning.

4.2 Abstract Concepts Maps

Results from an exit survey administered immediately after the PD's "Intro to Abstraction" lesson revealed teacher difficulty with the concept of abstraction. In response, the first author developed multiple additional mini supplemental abstraction lessons that modeled the use of abstraction to teach topics identified as problems of practice such as: "character", "theme", and "subtraction". Each mini lesson consisted of worked examples of abstraction and abstract concept maps and resulted in a succession of clarifying ideas regarding abstraction in elementary education.

Abstract concept maps were used to depict a consistent relationship structure between the elements of an abstraction, namely, the standard vertical relationship between an abstraction and the details from which it was derived. In each of the concept maps, the higher-level abstract concept appears at the top of the diagram, and its related lower-level details appear at the bottom of the diagram. An attempt was made to display two and only two layers of abstraction to keep representations of an abstraction as simple as possible.

4.3 "Character" Concept Map

The concept map for "character" (Fig. 1) was presented to educators in the following sequence:

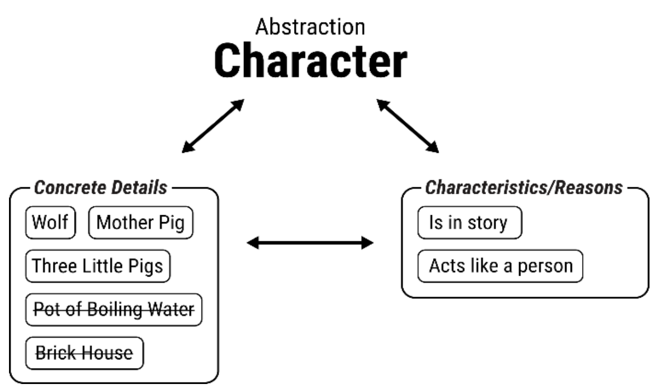

Figure 1: The "Character" concept map.

1. The word "character" was displayed first on a whiteboard and defined.
2. Potential concrete examples of "characters" from the story "The Three Pigs" were written down on the bottom-left.
3. Each potential concrete example of a "character" was then considered and given a reason for inclusion or exclusion.
4. All the reasons that some concrete details were included were written on the bottom-right side of the board.

5. Arrows were drawn linking the abstraction to its supporting concrete representations to the bottom-left and its characteristics to the bottom-right.
6. An arrow was drawn linking concrete representations with their shared characteristics to the right.

Analysis of the “character” concept map revealed the following clarifying ideas: *reverse abstraction*, an *evaluation step*, and a *pattern recognition step*.

4.3.1 Reverse Abstraction. In lower elementary education, students are not responsible for generating abstract concepts from empirical abstraction on their own. Instead, teachers typically give students definitions of new abstract concepts and help students develop their understanding of them. By starting with an abstraction first, students use empirical abstraction to *match* a given abstraction, essentially using a scaffolded abstraction process. Another perspective is that students would be engaging in *reverse* abstraction by taking a given abstraction and determining what concrete details support understanding of it. This slight but significant discrepancy in the way abstraction is used may be a source of confusion among elementary educators trying to envision how abstraction could be used in the classroom.

4.3.2 Evaluation Step. In the process of filtering details for an abstraction, students must *evaluate* details against the criteria for an abstraction. This evaluation step is an implicit part of the common CT definition of abstraction as the filtering of details. Being explicit about evaluation clarifies for primary educators the higher-order thinking [26] that students must perform while engaging in abstraction. Young students who may not have reached the developmental stage where they can critically evaluate their decisions [27] may need assistance doing so from teachers.

4.3.3 Pattern Recognition Step. Using the Rule of Three, where three is the minimum number of items required for a pattern, teachers can ask students to find commonalities between three concrete correct examples of an abstract concept, leading to potential development of understanding of the abstraction.

4.4 “Theme” Concept Map

The “theme” concept map (Fig 2.) was initially presented in the same way that the “character” concept map was presented, but after completing the middle row of supporting details, the first author noticed “Friendship”, “Loss”, and “Bravery” were abstract concepts themselves. To ensure understanding of those concepts, and by association understanding of the concept of “theme”, a second lower level of more concrete detail was added to the map. This created abstraction layers.

4.4.1 *Abstraction layers do exist in elementary education.* Potentially, multiple levels of a concept map might need to be generated until an appropriate level of concrete detail is exposed. Comparison to the “character” concept map shows the exponentially greater number of concrete details needed to support understanding of the top-level abstraction, especially if the Rule of Three is used.

4.3 “Subtraction” Concept Map

The “subtraction” concept map (Fig. 3) was developed in a small group session with two teachers, who took turns filling in the entire concept map. The map did not provide any clarifying ideas

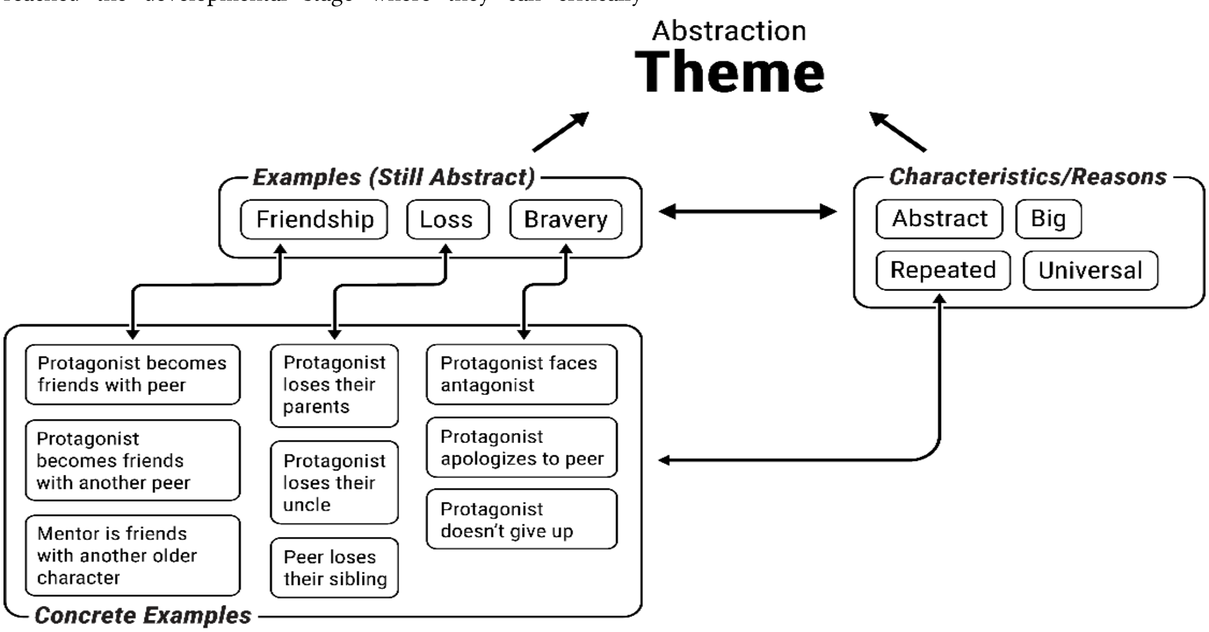

Figure 2: The multiple-level "Theme" concept map

about abstraction but confirmed that students try to understand abstract math concepts by looking at concrete versions of the abstraction [28]. One teacher stated that "I teach subtraction by describing eating cookies…I should try to find a couple more examples so students can compare examples."

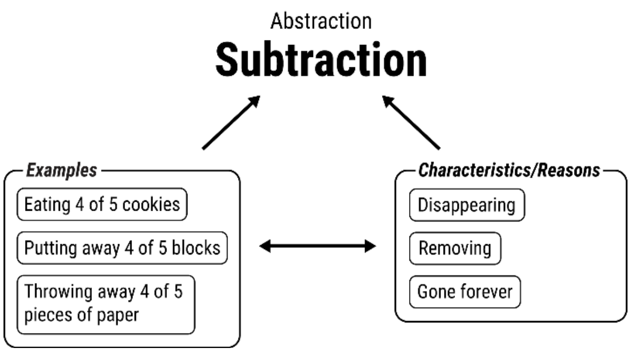

Figure 3: A math concept map

4.4 "Abstraction" Concept Map

Multiple examples of abstraction naturally lent themselves to being included in an abstract concept map of "abstraction" itself (Fig. 4). Unfortunately, the idea for this concept map was not conceived by the first author until after the PD had ended. However, at the time of the PD, one teacher did exclaim "We're abstracting abstraction!" after a fifth concept map was shown.

Applying pattern recognition to those concept maps revealed the following key details of abstraction in primary education:

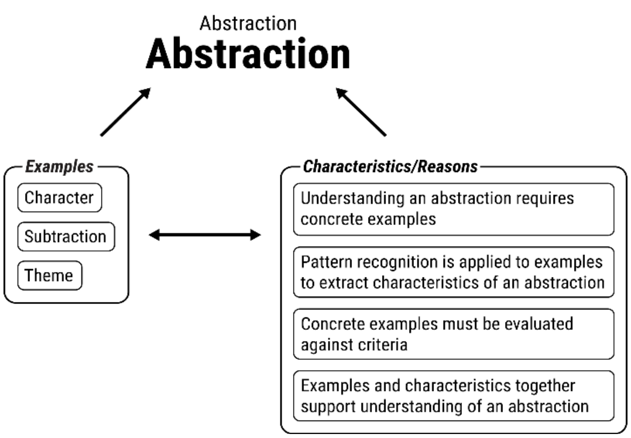

Figure 4: Abstracting abstraction

1. Understanding of an abstraction requires concrete examples.
2. Concrete examples must be evaluated against criteria.
3. By themselves, concrete examples only help so much. Pattern recognition must be applied *across multiple concrete examples* to extract and more deeply understand characteristics of an abstraction.
4. Examples and characteristics together support understanding of an abstraction.

5 TEACHER FEEDBACK

Immediately following the summer institute, a post survey captured teacher opinions on the abstraction lessons they received. Then, during the ensuing school year, written teacher reflections, which were completed after delivery of CT lessons, revealed teacher thoughts about abstraction lessons they designed and delivered. Preliminary analysis of the data broadly show support for the continued development of the abstraction pedagogy detailed in this paper and a growing teacher comfort with abstraction instruction in the classroom.

5.1 Evaluation Surveys

On the evaluation survey, teachers were asked (1) which of the CT concepts they found the most difficult to understand, (2) which of the four concepts they gained the most progress in, and (3) which of the four concepts they wanted to learn more about. For all three questions, abstraction was the most popular answer (Fig 5). Their responses align with literature reporting that teaching and grasping abstraction is challenging [2]. These results also acknowledge the potential effectiveness of concept maps for abstraction pedagogy and emphasize the necessity of offering additional opportunities to engage in learning this process.

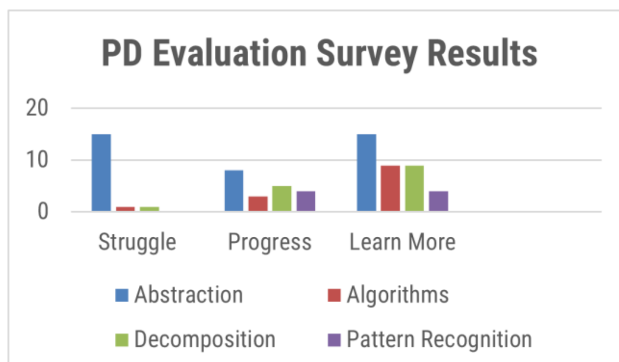

Figure 5: Teachers struggled with, made the most progress with, and wanted to learn the most about abstraction (n=20).

5.2 Lesson Reflections

After teaching a lesson on abstraction, one teacher reflected,

"Students were able to create their own Abstract personal narrative and knew what words to use. On the other hand, I felt like students struggled to understand abstraction…They understood the idea but could not relate the definition back to me.

This leads me to believe that they need to see more examples and continue to practice."

Echoing the need for practice, another teacher wrote, "I felt that the kids responded really well to the concept [of abstraction] and felt like experts by the end of the lesson. I think in future lessons we will need a lot of repetition and practice with the concept."

A third teacher described how abstraction contributed to understanding of core content, saying,

"…teaching idioms with abstraction worked really well together. I think it was valuable for the students to use abstraction in order to help them remember the meaning of the idioms (since idioms are so abstract :))…It helped create more meaningful conversation about the idioms."

Here we see teachers talking about abstraction, about how students use it in the classroom, how students are struggling with abstraction, and a potential way of alleviating that struggle. This conversation about abstraction shows a degree of comfort, one that suggests increasing teacher efficacy with abstraction.

6 DISCUSSION

6.1 Position Statement

Abstraction for elementary education should include the following activities:

1. *Identification of concrete examples and explanation of their relationship to a given abstraction.* Brainstorming for examples and explaining them affords students the opportunity to practice the evaluation step of abstraction.
2. *Pattern recognition across multiple concrete examples to extract characteristics that match those of a given abstraction.* Intentional use of pattern recognition across multiple concrete examples has the potential to quickly develop understanding. Conversely, regarding single examples in isolation may leave greater room for misconceptions.
3. For upper elementary students, *fully unscaffolded abstraction*, where students themselves are responsible for generating an abstraction.

These recommendations pervade the literature on abstraction, but their relevance remained unclear for our elementary educators until this process of abstracting abstraction highlighted their significance. Continued practice with *the parts of abstraction* described in this paper—explanation, evaluation, and pattern recognition—should help students make progress in understanding, applying, and explaining *the whole of abstraction.*

6.2 Impacts

Clarity at the elementary school level surrounding abstraction has a potentially outsized impact. With the amount of abstraction inherent in the elementary curriculum across the four core subjects, teachers have an extremely large number of opportunities to integrate abstraction into daily lessons, which would provide students with a significant amount of repeated practice with abstraction.

In computer science where abstraction permeates the field, students who have improved their abstraction skills will be able to develop and evaluate their programs and designs more accurately and efficiently, and potentially move between layers of abstraction more fluidly [8,9].

6.3 Limitations

The process of abstraction is often conducted over the course of many iterations. This project has reported on just one iteration of abstracting abstraction, and potential future iterations that include mini lessons on the process of abstraction in elementary education curricular activities such as summarizing, drawing conclusions, and modeling may result in the need to adjust recommendations.

The post-surveys used for the PD were designed prior to the design of the abstraction mini lessons. A more focused survey eliciting teacher feedback specifically about abstract concept maps would provide better data with which to evaluate the recommendations in this paper.

7 NEXT STEPS

Progress on this line of research will be conducted on multiple fronts. First, teachers will be observed in the classroom following these preliminary recommendations on abstraction instruction. Continued teacher feedback on the effectiveness of their abstraction lessons will inform potential adjustments to our recommendations. Second, additional mini lessons on abstraction will be designed in all core subjects across all elementary grade levels, which will either confirm the key aspects of the process of abstraction already identified or suggest needed changes. Lastly, assessment data on student use of abstraction will need to be collected to analyze the impact these recommendations have on student learning.

Eventually, researchers on this project hope to propose revisions to the definition of abstraction used in CT literature as well as identify a progression of abstraction learning for elementary education.

ACKNOWLEDGMENTS

This research was supported by the National Science Foundation (DRL-2219317), but the opinions expressed do not necessarily reflect the position, policy, or endorsement of the funding agency. We also gratefully acknowledge the teachers, administrators, and superintendents of the Ashland, Phoenix-Talent, and Lincoln School Districts in Oregon, who were instrumental in developing, providing feedback, and implementing this summer institute. Special thanks to Gianna McCardell for the graphic design of the abstract concept maps used in this paper.

REFERENCES

[1] Jeannette M Wing. 2006. Computational Thinking. *Commun. ACM* 49, 3 (2006), 33-35. https://doi.org/10.1145/1118178.1118215

[2] Yihuan Dong, Veronica Catete, Robin Jocius, Nicholas Lytle, Tiffany Barnes, Jennifer Albert, Deepti Joshi, Richard Robinson, Ashley Andrews. 2019. PRADA: A Practical Model for Integrating Computational Thinking in K-12 Education. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education (SIGCSE '19),* February 27 – March 2, 2019, Minneapolis, MN, USA. ACM Inc., New York, NY. 7 pages. https://doi.org/10.1145/3287324.3287431

[3] Kathryn M. Rich, Aman Yadav, Rachel A. Larimore. 2020. Teacher implementation profiles for integrating computational thinking into elementary mathematics and science instruction. *Educ Inf Technol (Dordr)* 25, 3 (Jul. 2020), 3161–3188. https://doi.org/10.1007/s10639-020-10115-5

[4] Eva Skuratowicz, Maggie Vanderberg, Eping E. Hung, Gladys Krause, Dominique Bradley, and Joseph P. Wilson. 2020. "I felt like we were actually going somewhere.". In *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education (SIGCSE '21),* March 13 - 20, 2021, Virtual Event, USA. ACM Inc., New York, NY. 7 pages. https://doi.og/10.1145/3408877.3432482

[5] Kevin P. Waterman, Lynn Goldsmith, and Marian Pasquale. 2020. Integrating computational thinking into elementary science curriculum: An examination of activities that support students' computational thinking in the service of disciplinary learning. *J. Sci. Educ.* 29, 1 (Feb. 2020), 53-64. https://doi.org/10.1007/s10956-019-09801-y

[6] Ndudi O. Ezeamuzie, Jessica S.C. Leung, and Fridolin S.T. Ting. 2022. Unleashing the potential of abstraction from cloud of computational thinking: A systematic review of literature. *J. Educ. Comput. Res.* 60, 4 (Jul. 2022), 877-905. https://doi.org/10.1177/07356331211055379

[7] Valerie Barr and Chris Stephenson. 2011.Bringing computational thinking to K-12: What is involved and what is the role of the computer science education community? *ACM Inroads* 2, 1 (Mar. 2011), 48-54. https://doi.org/10.1145/1929887.1929905

[8] Jeanette M. Wing. 2008. Computational thinking and thinking about computing. *Philos. Trans. Royal Soc. A* 366, 1881 (Oct. 2008), 3717-3725. https://doi.org/10.1098/rsta.2008

[9] Jeff Kramer. 2007. Is abstraction the key to computing? *Commun. ACM* 50, 4 (Apr. 2007), 36–42. https://doi.org/10.1145/1232743.1232745

[10] Yingxiao Qian and Ikseon Choi. 2023. Tracing the essence: Ways to develop abstraction in computational thinking. *Educ. Technol. Res. Dev.* 71, 3 (Jun. 2023), 1055–1078. https://doi.org/10.1007/s11423-022-10182-0

[11] Computational Thinking Competencies. 2023. Retrieved December 7, 2023 from International Society for Technology in Education: https://iste.org/standards/computational-thinking-competencies

[12] Tom Drummond. Tom Drummond Resources & Writings: Levels of Abstraction. Retrieved December 7, 2023 from https://tomdrummond.com/levels-of-abstraction-2/

[13] Damian Osborne. 2023. How to Draw an Abstract Drawing. Retrieved December 7, 2023 from https://www.damianosborne.com/how-to-draw-an-abstract-drawing

[14] CSTA, & ISTE. 2011. Operational Definition of Computational Thinking for K-12 Education. Retrieved December 7, 2023 from https://cdn.iste.org/www-root/Computational_Thinking_Operational_Definition_ISTE.pdf

[15] Orit Hazzan and Jeff Kramer. 2007. Abstraction in Computer Science & Software Engineering:. *System Design Frontier* 4, 1 (Jan. 2007), 6-14.

[16] Christoph Niemann. 2017. The Abstract-O-Meter. Retrieved December 7, 2023 from https://twitter.com/abstractsunday/status/891630379723096064

[17] David Statter and Michal Armoni. 2017. Learning abstraction in computer science. In *Proceedings of the 12th Workshop on Primary and Secondary Computing Education (WiPSCE '17),* November 8 - 10, 2017, Nijmegen, Netherlands. ACM Inc., New York, NY. 5-14. https://doi.org/10.1145/3137065.3137081

[18] Ibrahim Cetin and Ed Dubinsky. 2017. Reflective abstraction in computational thinking. *J. Math. Behav.* 47 (Sep. 2017), 70–80. https://doi.org/10.1016/j.jmathb.2017.06.004

[19] Evert W. Beth and Jean Piaget. 1974. *Mathematical epistemology and psychology*. Springer, Dordrecht, The Netherlands. https://doi.org/10.1007/978-94-017-2193-6

[20] Claudio MIROLO, Cruz IZU, Violetta LONATI, and Emanuele SCAPIN. 2021. Abstraction in computer science education: An overview. *Inform. Educ.* 20, 4 (Dec. 2021), 615–639. https://doi.org/10.15388/infedu.2021.27

[21] William Carlos Williams. 1938. The Red Wheelbarrow. In *The Collected Poems of William Carlos Williams, Volume I, 1909-1939, edited by A. Walton Litz and Christopher MacGowan.* New Directions Publishing Corporation, New York, NY (1986).

[22] T.R. Colburn and G.M. Shute. 2008. Metaphor in computer science. *J. Appl. Log.* 6, 4 (Dec. 2008), 526–533. https://doi.org/10.1016/j.jal.2008.09.005

[23] National Research Council. 2000. *How People Learn: Brain, Mind, Experience, and School: Expanded Edition.* The National Academies Press, Washington, DC. https://doi.org/10.17226/9853

[24] Orna Muller and Bruria Haberman. 2008. Supporting abstraction processes in problem. *Comput. Sci. Edu.* 18, 3 (2008), 187-212. https://doi.org/10.1080/08993400802332548

[25] Jacob Perrenet, Jan Friso Groote, and Eric Kaasenbrood. 2005. Exploring students' understanding of the concept of algorithm: levels of abstraction. *ACM SIGCSE Bull.* 37, 3 (Sep. 2005), 64-68. https://doi.org/10.1145/1151954.1067467

[26] Krathwohl, D. R. (2002). A revision of Bloom's taxonomy: An overview. *Theory into Practice, 41*(4), 212-218. doi:10.1207/s15430421tip4104_2

[27] McCleod, S. (2023). *Piaget's concrete operational stage of cognitive development.* Retrieved August 10, 2023, from simplypsychology.org: https://www.simplypsychology.org/concrete-operational.html

[28] Orit Hazzan. 2003. How students attempt to reduce abstraction in the learning of mathematics and in the learning of computer science. *Comput. Sci. Educ.* 13, 2 (2003), 95-122. https://doi.org/10.1076/csed.13.2.95.14202