International Society of the Learning Sciences

Promoting Computational Thinking Through Science-Engineering Integration Using Computational Modeling

Satabdi Basu, SRI International, satabdi.basu@sri.com
Kevin W. McElhaney, Digital Promise, kmcelhaney@digitalpromise.org
Arif Rachmatullah, SRI International, arif.rachmatullah@sri.com
Nicole Hutchins, Vanderbilt University, nicole.m.hutchins@vanderbilt.edu
Gautam Biswas, Vanderbilt University, gautam.biswas@vanderbilt.edu
Jennifer Chiu, University of Virginia, jlc4dz@virginia.edu

Abstract: Careers in science, technology, engineering, and mathematics (STEM) increasingly rely on computational thinking (CT) to explore scientific processes and apply scientific knowledge to the solution of real-world problems. Integrating CT with science and engineering also helps broaden participation in computing for students who otherwise would not have access to CT learning. Using a set of emergent design guidelines for scaffolding integrated STEM and CT curricular experiences, we designed the Water Runoff Challenge (WRC) - a three-week unit that integrates Earth science, engineering, and CT. We implemented the WRC with 99 sixth grade students and analyzed students' learning artifacts and pre/post assessments to characterize students' learning process in the WRC. We use a vignette to illustrate how anchoring CT tasks to STEM contexts supported CT learning for a student with low prior CT proficiency.

Introduction

Preparing pre-college students for careers in science, technology, engineering, and mathematics (STEM) requires them to engage in authentic STEM practices. These practices increasingly rely on computational thinking (CT) to explore scientific processes and apply scientific knowledge to design solutions for real-world problems (Freeman et al., 2014). The emergence of computation as the third pillar (alongside theory and experiment) of science and engineering and the inclusion of CT in the U.S. *Framework for K-12 Science Education* require that primary and secondary schools incorporate CT into existing STEM curricula (National Research Council [NRC], 2012). Integrating CT with required STEM subjects also helps broaden participation in computing for students who otherwise would not have access to CT learning.

A computational model (CM) is a mathematical representation of a system created using a formal notation (e.g., a programming language) so that the system behavior can be studied using computer simulations (Melnik, 2015). The collective practices of developing, testing, refining, and using CMs (National Research Council, 2012)—can be the central link between students' scientific understanding of a system and engineering a problem solution related to the system, especially for systems that cannot be easily investigated using physical materials. This approach merges the benefits of developing computational artifacts with simulation-based learning and problem solving.

This paper examines how CT can be promoted through science-engineering integration using computational modeling, particularly for younger students with no or limited prior exposure to computing. Recent research on how students develop CMs of science phenomena has mostly focused on integration of CT with either science (e.g., Waterman et al., 2020) or engineering (e.g., Magana & de Jong, 2018) but not both. Introducing CMs for elementary school students presents challenges such as students' limited mathematics proficiency, class time needed to develop programming/modeling skills, and the training needed for elementary teachers who are typically generalists and may not have deep CT, engineering, and/or science expertise. We describe curriculum design guidelines that can promote integration of CT with science and engineering via computational modeling and report on the implementation of an integrated STEM+CT unit designed for fifth and sixth grade students.

Theoretical and Empirical Perspectives

Synergistic learning of STEM and computing

Past research on synergistic STEM and computing learning (e.g., Hutchins et al., 2020; Sengupta et al., 2013; Waterman et al., 2020) has shown that computational modeling can deepen students' science learning. Complex scientific systems become easier to understand when the system structure and behaviors are explained computationally. CMs provide explicit mechanisms for constructing and visualizing scientific phenomena and reasoning about scientific processes as a sequence of step-by-step changes as opposed to creating aggregated mathematical models (Hutchins et al., 2020). Learning to program CMs involves learning how to formally articulate mechanisms and explanations, a core scientific practice. In turn, programming also becomes easier to learn when anchored in real-world problem contexts (Sengupta et al., 2013). Alongside documenting the benefits of computational modeling, research has also identified that students may have significant difficulties with developing scientific CMs. Students may struggle with both understanding the science concepts underlying CMs and representing system variable relationships using computational expressions (Sengupta et al., 2013). Students may also have trouble generalizing or abstracting the scientific processes into variables and expressions (Basu et al., 2016). These findings indicate a need to scaffold students' computational modeling experience as they grapple with the multi-fold complexities of scientific knowledge, modeling practices, and computational representations.

Framework for integrating science, engineering, and CT

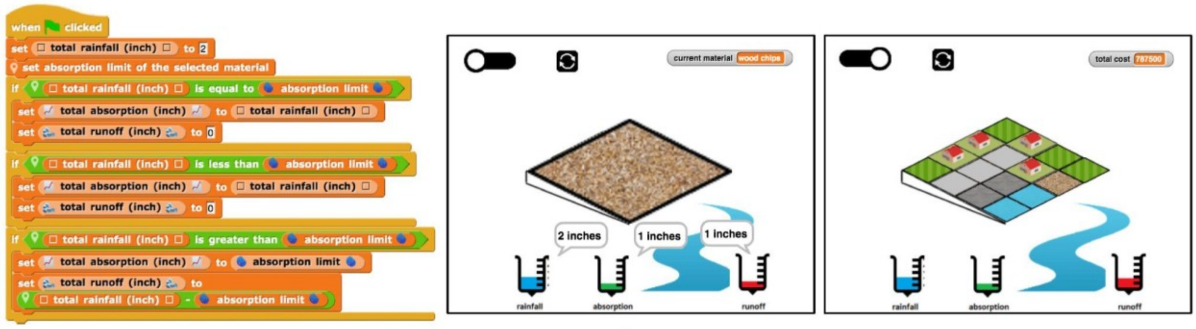
Relatively little research has examined how CMs can enable the developing or testing of engineering solutions related to science phenomena. We integrate science, engineering and CT via computational modeling by leveraging a design framework (McElhaney et al., 2020) that involves (1) defining an engineering design challenge whose solution is governed by an underlying science phenomenon, (2) supporting students to develop a CM of the science phenomenon, and (3) supporting students to use their CMs to design, test, and refine solutions to the engineering challenge. The framework integrates research on designing computational environments that enable students to model science phenomena (e.g., Hutchins et al., 2020; Sengupta et al., 2013), and research investigating how students use simulations to engage in problem-solving (e.g., Magana & de Jong, 2018; Weintrop et al., 2016). To implement this framework, we leveraged the following design perspectives to integrate science, engineering, and CT. These perspectives guide the design of the Water Runoff Challenge (WRC), the integrated STEM+CT unit used for this study (described in further detail in the Methods section).

- **Create an explicit need for computational modeling.** To achieve authenticity in the design of a STEM+CT curriculum unit, the overarching design challenges should make the use of computational tools consequential. Students should be able to recognize the unique affordances of CMs for solving STEM problems, making scientific predictions, and designing engineering solutions (Weintrop et al., 2016). This design guideline also promotes the CS practice of Recognizing and Defining Computational Problems.
- **Maintain coherence across system representations.** In a STEM+CT unit, students must interact with and/or generate multiple system representations (e.g., pictorial, mathematical, narrative, and computational) of the science phenomenon being investigated. In order to work together to promote integrated learning, these representations need to be conceptually coherent (Ainsworth, 1999) and consistent with respect to the system variables and their relationships and the computational and scientific terminology employed.
- **Employ a Domain Specific Modeling Language (DSML).** DSMLs reduce the challenges that learners face when modeling scientific scenarios by adapting the level of system abstraction to conceptual levels appropriate for learners (Hutchins et al., 2020). Relative to using a general-purpose programming language to develop scientific models, students can focus on the relevant science concepts and system variables without having to attend to the program syntax. The use of DSMLs for computational modeling reduces the cognitive load associated with learning to program and shifts the focus toward science modeling.

Methods

Designing to promote CT learning in integrated STEM+CT instruction

In activities integrating science, CT and engineering, whether learning CT will promote or interfere with science and engineering learning depends on the learner's ability to manage the challenges of disciplinary integration. The ability to successfully leverage synergies between computing and STEM learning requires scaffolding, especially for younger students. We describe a set of six design guidelines to scaffold young students and improve their understanding of CT with specific examples from the WRC, a multi-week unit that integrates CT with Earth science and engineering for fifth and sixth grade students (Zhang et al., 2020). The WRC uses the design perspectives described above to engage students with the problem of a schoolyard that floods during periods of heavy rainfall. Students are challenged to resurface the schoolyard to minimize the water runoff while meeting design constraints on cost, accessibility for students with physical disabilities, and practical use. Based on the science principle of matter conservation, students create pictorial models explaining how a part of the rainfall is absorbed into the ground based on the surface material properties, while the rest remains on the surface and flows downhill (runoff), resulting in flooding of nearby areas. Students then use a block-based DSML to develop a corresponding CM (Figure 1a) for a unit surface area (one square). The DSML used in the WRC includes domain-specific constructs such as rainfall, water absorption, runoff, and surface material, as well as CT constructs that are needed in the runoff CM. This CM determines the amount of water runoff based on the amount of rainfall and the (predefined) absorption limit of the surface material. As students build the CM, they test it using different values of rainfall and different surface materials (Figure 1b). After students develop a working CM, their model code is then incorporated into a simulation environment comprising 16 squares for the engineering design task (Figure 1c) where students need to configure a set of appropriate materials for resurfacing the schoolyard.



(a) (b) (c)

Figure 1. *(a) A Working Runoff CM (b) Single Square CM Testing (c) 16-Square Engineering Design*

Below, we synthesize the WRC unit design features into a set of emergent design guidelines encompassing three categories of support for promoting CT in integrated STEM+CT instruction. These design guidelines have emerged from iterative implementation of the WRC unit for fifth and sixth grade students and its refinement using insights from classroom studies (McElhaney et al., 2020; Zhang et al., 2020).

Preparing Students for CM Development

1. **Support conceptual modeling of science phenomena prior to CM development.** In order to successfully develop a CM, students first need a strong scientific and mathematical understanding of the system behavior, which they can then express using CT constructs in a computational modeling representation. Having students first develop a conceptual model of the science phenomenon helps learners establish a firm scientific understanding of system behavior and variable relationships before representing the system computationally using the same variable names and relationships. In the WRC unit, students engage in a series of conceptual modeling activities: (1) a hands-on activity to observe differences in absorption characteristics among surface materials; (2) a pictorial modeling activity (Figure 2a) where students illustrate relationships among rainfall, absorption, and runoff; and (3) articulation of mathematical rules that express water runoff in terms of rainfall and material absorption limits.
2. **Introduce CT concepts required for developing CMs using a familiar context**. In addition to science understanding, students must have adequate proficiency with CT concepts and practices and an understanding of the computational modeling language in order to develop a CM. To design broadly accessible computational modeling experiences for fifth and sixth grade students, one cannot assume that students have prior familiarity with programming or CT. Hence, activities should engage students in CT concepts and practices required for developing the target CM by first presenting them in a familiar (non-scientific) context and in an isomorphic way to the CM. For example, in the WRC unit, we developed unplugged activities that engage students in a series of dice games of increasing complexity before students develop the runoff CM. The dice games introduce students to the concepts of conditionals, variables and expressions by assigning players different scores based on their dice values (Figure 2b).

Supporting CM development

3. **Engage a limited and age-appropriate set of CT concepts and practices**. When engaging fifth and sixth grade students who have no assumed prior knowledge of CT in computational modeling of science phenomena, curriculum units need to limit students' engagement to a small set of CT concepts and practices that are needed to model the phenomenon in a compelling fashion and can be learned with a limited amount of classroom instruction. Because science and CT learning are so tightly integrated in such units, limiting the target CT concepts may also require simplification of the science concepts and system representation by limiting the number of system components and abstracting their relations. In the WRC unit, we target the CT concepts of sequences, variables, expressions, and conditionals (Figure 1a) and the practices of developing, testing, and debugging computational artifacts, as well as using computational artifacts for problem-solving. We arrived at this target CM after several previous implementation iterations that involved several temporal and rate-based variables and the CT concept of repeated conditionals. Although scientific CMs are typically temporal in nature, the temporal runoff CM turned out to be too complex for typical fifth and sixth grade

students. Hence, we omitted the CT concept of iteration from the current version of the CM to make it mathematically and computationally tractable for young students.

4. **Support model decomposition**. For fifth and sixth grade students using a programming environment for the first time, it can be helpful to decompose the CM development process into conceptually meaningful chunks. For example, in the WRC unit, because the system behavior is modeled according to three rules (rainfall less than, equal to, or greater than the surface absorption limit), we designed one classroom activity for students to program each rule. Each activity guides students through an incremental process of prediction, model development, testing, debugging, and reflection by adding the code for one rule at a time.

Supporting CM Evaluation

5. **Support CM testing and the selection of test cases.** Learners need support on CT practices such as systematically and thoroughly testing CMs and identifying appropriate test cases for CM evaluation. Our initial studies revealed that selecting appropriate test cases is not trivial for fifth and sixth grade students. In the WRC unit, students are prompted to use three test cases corresponding to the three conditional rules in the CM (Figure 2c) and to test their CMs with at least two different surface materials. Students model the CM in parts over three class periods and document outcomes with all three test cases each day.
6. **Include opportunities for students to engage with debugging**. Debugging CMs goes beyond merely identifying and fixing errors in code; it integrates CT and science by requiring learners to determine whether the CM correctly represents the underlying science model. Debugging CMs involves identifying and correcting computational errors as well as errors stemming from incorrect representations of the underlying science content. Embedding explicit debugging opportunities in STEM+C units can help shed light on causes of student challenges during CM development. In the WRC unit, we started some CM development lessons with a short debugging task. Tasks were based on code students had written the previous day and prompted reflection on previous coding activity. We also embedded debugging tasks as part of formative assessments (e.g., Figure 3) that students could complete in class or at home.

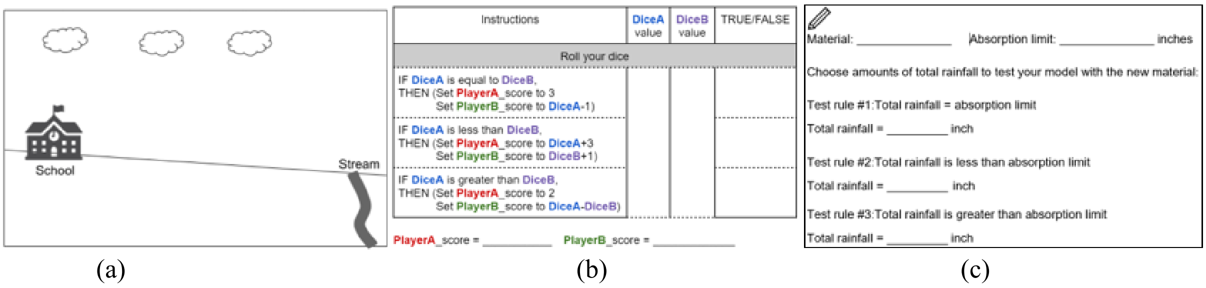


Figure 2. *(a) A pictorial conceptual modeling task, (b) A dice game CT task, (c) A test case selection task*

Classroom study using the WRC unit

We conducted a classroom study with 99 sixth grade students in the southeastern United States. Students completed the WRC unit over 15 class periods in the Fall of 2019, with identical pre-post tests administered before and after the intervention. Students worked for 45 min per day, three days per week during their regular science classes, and 75 min, twice a week, with additional personalized-learning time. Some students had prior programming experience with Scratch, while others had no previous programming experience. The participating teachers were highly experienced with teaching science and received four days of training before the study.

Data sources, scoring, and analysis

We examined four primary types of student data: pre-post assessment performance, final computational models, responses to activities in student workbooks, and responses to formative assessment tasks.

Students completed a paper-and-pencil **pre-post assessment** that was split into a science and engineering component and a CT component (McElhaney et al., 2019). Science/engineering tasks were aligned with several Next Generation Science Standards (NGSS) Performance Expectations in Earth science and engineering (NGSS Lead States, 2013). Of the five science/engineering tasks, three asked students to apply an engineering practice to the solution of a water runoff problem, one asked students to develop a model of water runoff in a context different from the WRC, and one asked students to apply an engineering practice to the solution of a problem unrelated to water runoff. For example, in one task, students compare two street designs based on criteria related to water runoff performance, usage, and cost. The science/engineering task rubrics measured the extent to which students could apply the focal science/engineering practice to the water runoff context and/or other engineering design criteria. The CT assessment tasks were aligned with the CT concepts and practices addressed in the runoff CM.

Students could score a maximum of 23 points on science and engineering and 13 points on CT. The rubrics used for coding and scoring the tasks were updated from our previous work. Two researchers received 5 hours of training on the rubrics, graded 5% of the test submissions (randomly selected) together to establish initial grading consistency, and then graded another 20% to establish inter-rater reliability (Cohen's κ at ≥ 0.8 level on all items). All differences in the coding were discussed and resolved before the remaining 75% of test submissions were graded by a single researcher.

Students' **final runoff CMs** were logged and scored using a rubric targeting conservation of matter rules and conditional statements for the different rules. The rubric rewarded four main criteria of the CMs: (1) assigned appropriate values to the total rainfall and absorption limit variables, (2) included code for three conditional statements that compared the total rainfall and absorption limit variables, (3) updated the absorption and runoff variables correctly for each of the three conditions, and (4) code is generalizable (e.g., assigned variables to expressions rather than to hardcoded numeric values).

Students completed a paper-and-pencil **workbook** over the course of the unit. We scored and analyzed student responses to specific workbook activities such as the conceptual modeling tasks, dice game activities, test case identification tasks, and a CM debugging task. A series of four student **conceptual models** and corresponding written explanations were scored based on their adherence to the conservation of matter principle and articulation of the causal relations governing the flow of rainwater. We also scored student performance on the **dice games** that introduced students to CT skills. Each of the four games were scored based on whether students could correctly evaluate conditional statements and interpret variable assignment statements. We report dice game scores only for students who documented outcomes from all four games ($n = 62$). We scored students' **choice of test cases** for testing the three rules in the runoff CM for a material of their choice based on whether the test cases address each of the model's three rainfall conditions. We scored a **debugging** task where the variable assignment for the absorption and runoff variables were flipped for a particular condition. Finally, we scored student responses to three select **formative assessment tasks** that assessed students' ability to predict the output of given CMs, identify errors, and fix buggy CMs.

We performed paired *t*-tests to measure students' pre-to-post learning on science, engineering, and CT. We also examined correlations among students' pre-post performance and performance on various CM development and evaluation activities to better understand students' learning process in the WRC unit.

Findings

Overall STEM and CT learning gains

Students' science/engineering scores improved significantly with a high effect size ($p < .0001$, $d = 0.81$) from an average of 13.3 (SD=2.8) points at pre-test to an average of 15.6 (SD=2.9) points at post-test. Students' CT scores also improved significantly with a high effect size ($p < .001$, $d = 0.83$) from an average of 6.2 (SD=2.6) points at pre-test to an average of 8.4 (SD=2.7) points at post-test. These pre-post changes reflect improvement in students' ability to develop scientific models, define engineering problems and design criteria, and engage with the CS concepts of variables, expressions, and conditionals. Below, we describe relations among students' pre-post changes and performance on curricular activities, and illustrate how students' science/engineering proficiency may be supporting students' CT learning using a vignette.

Student interactions with the computational modeling scaffolds in the WRC

Students demonstrated large variation in their CM construction behaviors, but about two-thirds of the students finally constructed a correct implementation of the runoff CM with instructional support. The average CM score was 13.6 (SD=2.5) out of 15. For students who were not able to construct a correct CM implementation, common challenges included omitting conditional statements (resulting in not covering all comparisons between the rainfall and absorption limit variables), not assigning values to both absorption and runoff variables in each of the conditions, and not assigning variables to correct values or assigning variables to hardcoded values. In particular, many students found it challenging to express the relation "set total runoff to (total rainfall − absorption limit)" when "total rainfall is greater than absorption limit."

These challenges with CM development align with challenges observed when students worked on the conceptual modeling and unplugged dice game activities prior to the runoff CM development. While both these activities prepare students with the science and CT understanding needed to create the runoff CM, both activity performances have small, non-significant correlations with students' CM scores, indicating that the relations may not be linear. Students scored an average of 27.0 (SD=4.5) points out of a possible 32 on the conceptual modeling tasks. Students were generally able to conceptually model the conservation principle and numerically predict the values of absorption and runoff given values of rainfall and absorption limit. However, many students struggled to provide mechanistic explanations for their numeric predictions or articulate generalizable rules to express the system variable relationships. On the dice game activities, 36 students earned the maximum total score of 8 on all four dice games. Students scored an average of 7.2 points (*SD*=1.3). Students were least successful with the fourth dice game, which was designed to be the most difficult - it involved evaluating conditional statements comparing two variables and interpreting variable values assigned to other variables and expressions. Most of the challenges stemmed from interpreting the variable assignment statements correctly.

We observed wide variation in students' performance on the activities designed to scaffold CM evaluation (selecting test cases and debugging). As we expected, model evaluation practices, in particular debugging, were challenging for sixth grade students. Students scored an average of 3.5 (*SD*=2.2) out of 6 points on the test case selection activity and an average of 8.0 (*SD*=3.0) out of 22 points on the debugging activities. Students' debugging performance was significantly correlated with their conceptual modeling scores (ρ= .27, p< .01) and CM scores (ρ= .23, p < .05). This result highlights the nature of debugging CMs in STEM+CT units, which entails both determining the accuracy of the underlying science model and its computational representation. Some common debugging challenges included: (a) the "superbug" challenge (Pea, 1986) where students thought the computer has a mind of its own and will always generate the correct result, (b) identifying and correcting all errors when the CM comprised multiple bugs, (c) believing that variable values cannot be re-assigned once assigned to initial values, and (d) incorrectly switching operands of logical operators and assignment statements.

Relations among computational modeling and pre-post performance

We conducted a correlational analysis among the pretest and posttest scores for science/engineering and CT and students' performance on the learning artifacts described above (see Table 1 for correlation coefficients and significance values). The significant correlations between (i) conceptual modeling and post-science/engineering, (ii) unplugged dice games and post-CT, and (iii) computational modeling and post-CT are consistent with expectations based on the alignment with their respective disciplines. The significant correlation between test cases and post-CT scores reflects how test cases can engage CT constructs such as variables and expressions, conditional logic, and code comprehension in a science modeling context. The significant correlations between students' debugging performance and posttest scores in both science/engineering and CT further illustrates the extent to which debugging a CM is an integrated activity—debugging a CM entails evaluation of both the code itself as well as the underlying scientific system model.

Table 1. *Correlations among pre-scores, post-scores and curricular activity performance (** p < .01, * p <.05)*

	Pre - science & engineering	Pre - CT	Conceptual modeling	Unplugged dice games	Computational modeling	Selecting test cases	Debugging
Post - science & engineering	.390**	.064	.247*	.127	.061	.176	.251*
Post - CT	.432**	.609**	.125	.319*	.215*	.206*	.310*

Finally, we note that there was a higher correlation between pre-science/engineering scores and post CT scores than between pre- and post-science/engineering scores. This somewhat unexpected finding may reflect the extent to which students' science/engineering knowledge and skill could be "bootstrapping" CT learning by enabling students to leverage STEM concepts and practices toward the advancement of their CT proficiency. We illustrate examples of how the integrated performance tasks may be enabling students' science/engineering proficiency to support their CT learning in a vignette below.

Improved CT through science & engineering integration and CM scaffolds

We highlight the work of a particular student, Taylor (pseudonym), who illustrates how science/engineering proficiency may be supporting CT learning across the STEM+CT integrated computational modeling activities in the WRC unit. Taylor achieved a high pre-science/engineering score of 17.5 (93rd percentile) and maintained the same score on the posttest (67th percentile). They achieved a low pre-CT score of 4.5 (25th percentile) and improved to the maximum post-CT score of 13. The pretest scores suggest that at the beginning of the study, Taylor was a proficient science student with little previous exposure to programming.

Despite achieving a low pre-CT score, Taylor exhibited a particularly high level of performance on activities associated with post-CT performance. Taylor earned the maximum score of 8 points on the dice game activities, achieved by only about one third of the students in the study. Taylor also exhibited a strong conceptual understanding of test cases, earning 5 out of a possible 6 points on the test case selection task and with the only error being arithmetic in nature (incorrectly transforming the number 0.09 to 0.9 prior to performing the test). Taylor's strongest performance was in the difficult debugging tasks, earning 14 points (the second highest among all students in the study). For instance, in response to the task shown in Figure 3(a), Taylor identified both instances where the system variables were incorrectly reversed, writing "In many occurrences (sic), Sumi needs to switch the Total Rainfall and Total Absorption twice on the second IF THEN statement." In response to the task in Figure 3(b), Taylor correctly predicted that the code in the nested 'if' block would not be executed for an absorption limit of 0.8 inches and correctly articulated that moving the nested 'if' block to the end of the program would debug the model. Taylor's responses to both the test case activities and debugging activities illustrate how designing CT tasks to be embedded in STEM contexts have the potential to help students leverage STEM understanding toward advancing their CT proficiency (including programming skill).

(a) (b)

Figure 3. Formative debugging task assessing (a) variable assignments and conditional expressions and (b) code sequencing and control flow in conditional structure

Discussion and Conclusion

This paper illustrates how CT learning can be promoted for upper-elementary school students through the coherent integration of science, engineering, and computational modeling. We describe activities such as conceptual modeling, unplugged dice games, test case identification, and CM debugging that support the integration of STEM and CT disciplines and discuss how students' engagement with them may be associated with learning in STEM and CT disciplines. These activities enable students having no prior programming experience to successfully develop CMs with minimal programming instruction and offer the potential for students to use their science and engineering knowledge to bootstrap their CT proficiency. Though students' pretest and posttest scores were strongly correlated, the design guidelines related to CM development and evaluation informed scaffolding for students' CT learning even when students started with low CT proficiency.

Our design for promoting CT for fifth and sixth grade students using computational modeling in an integrated science-engineering context also contributes to ongoing research on the nature and extent of CT integration and the feasibility and usefulness of programming and deep CT integration for upper-elementary students and teachers. Analyses of elementary teachers' plans for integrating CT in science lessons (Coenraad et al., 2022) reveal the use of computer simulations and data as some of the most common CT practices integrated while programming is one of the least common practices. When programming is used, it is often used to exhibit science knowledge gained through other means rather than further science learning. Overall, the integration is often limited to the bottom two levels of the CT integration framework (Coenraad et al., 2022) - *exist* (identify science practices that also count as CT) and *enhance* (use small CT activities to support science learning). This finding highlights the importance of presenting teachers with usable models of deeper CT integration that include a range of CT practices. The WRC demonstrates the feasibility of a science and engineering curriculum for fifth and sixth grade students involving numerous CT practices (computational simulations, programming, systems thinking, using data) and a deep level of CT integration where CT and science learning are mutually supportive.

Further, our findings add to a growing body of research examining the relationship between learning in CT and STEM disciplines (Lee at al., 2020). The framing of CT as a science and engineering practice in the US Science Education Framework, as well as schools' typical implementation of STEM as core subjects into which CT can be inserted, position CT as domain-general skills that advance STEM learning and prepare students for success across STEM disciplines. Our findings provide preliminary evidence of the converse relationship, where STEM disciplinary proficiency supports CT learning. Additional research studies are needed to clarify the mechanisms by which CT and STEM proficiency support each other. For instance, our findings align with studies (e.g., Hutchins et al, 2020) that identify debugging as an especially important connection between CT and scientific modeling.

References

Ainsworth, S. (2006). DeFT: A conceptual framework for considering learning with multiple representations. *Learning and Instruction*, *16*(3), 183-198.

Basu, S., Biswas, G., Sengupta, P., Dickes, A., Kinnebrew, J. S., & Clark, D. (2016). Identifying middle school students' challenges in computational thinking-based science learning. *Research and Practice in Technology Enhanced Learning*, *11*(1), 1-35.

Coenraad, M., Cabrera, L., Killen, H., Plane, J., & Ketelhut, D. J. (2022). Computational thinking integration in elementary teachers' science lesson plans. In Computational Thinking in PreK-5: Empirical Evidence for Integration and Future Directions (pp. 11-18).

Freeman, S., Eddy, S. L., McDonough, M., Smith, M. K., Okoroafor, N., Jordt, H., & Wenderoth, M. P. (2014). Active learning increases student performance in science, engineering, and mathematics. *Proceedings of the National Academy of Sciences*, *111*(23), 8410-8415.

Hutchins, N. M., Biswas, G., Maróti, M., Lédeczi, Á., Grover, S., Wolf, R., Blair, K. P., Chin, D., Conlin, L, Basu, S., & McElhaney, K. (2020). C2STEM: A system for synergistic learning of physics and computational thinking. *Journal of Science Education and Technology*, *29*(1), 83-100.

Hutchins, N. M., Biswas, G., Zhang, N., Snyder, C., Lédeczi, Á., & Maróti, M. (2020). Domain-specific modeling languages in computer-based learning environments: A systematic approach to support science learning through computational modeling. *International Journal of Artificial Intelligence in Education*, *30*(4), 537-580.

Lee, I., Grover, S., Martin, F., Pillai, S., & Malyn-Smith, J. (2020). Computational thinking from a disciplinary perspective: Integrating computational thinking in K-12 science, technology, engineering, and mathematics education. Journal of Science Education and Technology, 29(1), 1–8.

Magana, A. J., & de Jong, T. (2018). Modeling and simulation practices in engineering education. *Computer Applications in Engineering Education*, *26*(4), 731-738.

McElhaney, K. W., Zhang, N., Basu, S., McBride, E., Biswas, G., & Chiu, J. L. (2020, January). Using computational modeling to integrate science and engineering curricular activities. In In M. Gresalfi & IS Horn (Eds.). The Interdisciplinarity of the Learning Sciences, 14th International Conference of the Learning Sciences (ICLS) 2020 (Vol. 3).

McElhaney, K.W., Basu, S., Wetzel, T., & Boyce, J. (2019, April). *Three-Dimensional Assessment of NGSS Upper Elementary Engineering Design Performance Expectations*. Paper presented at the NARST Annual International Conference, Baltimore, MD.

Melnik, R. (Ed.). (2015). *Mathematical and computational modeling: with applications in natural and social sciences, engineering, and the arts*. John Wiley & Sons.

National Research Council. (2012). *A framework for K-12 science education: Practices, crosscutting concepts, and core ideas*. National Academies Press.

NGSS Lead States. (2013). *Next Generation Science Standards: For states, by states*. Washington, DC: National Academies Press.

Pea, R. D. (1986). Language-independent conceptual "bugs" in novice programming. Journal of educational computing research, 2(1), 25-36.

Sengupta, P., Kinnebrew, J. S., Basu, S., Biswas, G., & Clark, D. (2013). Integrating computational thinking with K-12 science education using agent-based computation: A theoretical framework. *Education and Information Technologies*, *18*(2), 351-380.

Waterman, K. P., Goldsmith, L., & Pasquale, M. (2020). Integrating computational thinking into elementary science curriculum: An examination of activities that support students' computational thinking in the service of disciplinary learning. *Journal of Science Education and Technology*, *29*(1), 53-64.

Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining computational thinking for mathematics and science classrooms. Journal of Science Education and Technology, 25(1), 127-147.

Zhang, N., Biswas, G., McElhaney, K. W., Basu, S., McBride, E., & Chiu, J. L. (2020, July). Studying the Interactions Between Science, Engineering, and Computational Thinking in a Learning-by-Modeling Environment. In International Conference on Artificial Intelligence in Education (pp. 598-609).

Acknowledgments

We gratefully acknowledge the support of National Science Foundation award DRL-1742195. We also appreciate the contributions of our project colleagues and the suggestions of two anonymous reviewers. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.