Using Computational Modeling to Integrate Science and Engineering Curricular Activities

Kevin W. McElhaney, Digital Promise, kmcelhaney@digitalpromise.org
Ningyu Zhang, Vanderbilt University, ningyu.zhang@vanderbilt.edu
Satabdi Basu, SRI International, satabdi.basu@sri.com
Elizabeth McBride, SRI International, beth.mcbride@sri.com
Gautam Biswas, Vanderbilt University, gautam.biswas@vanderbilt.edu
Jennifer L. Chiu, University of Virginia, jlchiu@virginia.edu

Abstract: We articulate a framework for using computational modeling to coherently integrate the design of science and engineering curricular experiences. We describe how this framework informs the design of the Water Runoff Challenge (WRC), a multi-week curriculum unit and modeling environment that integrates Earth science, engineering, and computational modeling for upper elementary and lower middle school students. In the WRC, students develop conceptual and computational models of surface water runoff, then use simulations incorporating their models to develop, test, and optimize solutions to the runoff problem. We conducted a classroom pilot study where we collected students' learning artifacts and data logged from their use of the computational environment. We illustrate opportunities students had to integrate science, engineering, and computational thinking during the unit in a pair of contrasting vignettes.

Keywords: science, engineering, computational thinking, curriculum design, elementary

Introduction

The emergence of computation as the third pillar (alongside theory and experiment) of science and engineering requires schools to incorporate computational thinking (CT) into the K-12 curriculum. Consequently, students need opportunities to learn the CT skills that will enable them to contribute to scientific discoveries, engineering solutions, and computing innovations (Wing, 2008). One place where deep connections among CT, science, and engineering can be strongly featured is in the practice of modeling. In science and engineering, modeling involves selecting aspects of a phenomenon to be modeled, identifying relevant variables and their relations, developing formal representations, implementing the relations using the formal representation language, and analyzing system representations to predict system behavior and achieve problem solutions (e.g., Basu, et al., 2012; Lehrer & Schauble, 2006). Relevant aspects of CT that align with the practice of modeling include development of computational artifacts, abstraction, problem decomposition, and debugging.

Recent research has produced successful activities where learners develop computational models of science phenomena (e.g., Basu, Biswas, & Kinnebrew, 2017; Wilkerson-Jerde, Wagh, & Wilensky, 2015), but relatively few that also ask learners to develop, test, and refine engineering solutions related to those phenomena. Connecting both engineering and science with computational modeling provides opportunities to foster student learning of all three disciplines (NRC, 2011). Computational models address numerous shortcomings of physical materials for engaging in engineering activities by emphasizing engineering as an approach to problem-solving (rather than as building physical artifacts). Computational models also support the modeling of complex systems and data-driven iterative refinement of designs. In these ways, computational modeling is particularly well suited to both explaining scientific phenomena and developing engineering solutions related to those phenomena.

This paper addresses the question: How can curriculum materials use computational modeling to coherently integrate the science and engineering disciplines? We describe the design approach for the Water Runoff Challenge (WRC) (Chiu et al., 2019), a multi-week curriculum unit that integrates Earth science, engineering, and CT for upper elementary and lower middle school students. We illustrate opportunities students had to integrate the disciplines during the unit using a pair of contrasting vignettes from a classroom pilot study.

Design perspectives

Our design approach integrates numerous perspectives on the design of STEM and computing curriculum materials and learning technologies.

- **Evidence-centered design.** We use evidence-centered design (Mislevy & Haertel, 2006) to systematically unpack the target science, engineering, and computing concepts and practices and specify

anchor points for curriculum and technology design. To guide design, we articulate a set of *learning performances* (Harris, Krajcik, Pellegrino, DeBarger, 2019; Krajcik, McNeill, & Reiser, 2008), which are learning goals that integrate disciplinary content knowledge, disciplinary practices, and crosscutting content that students need to attain as they progress toward broad learning goals. Learning performances integrate smaller aspects of performance across disciplines, thereby providing specific targets for the design of curricular activities and helping identify associated formative assessment opportunities.

- **Integrating science and engineering.** Our work extends research and design that integrate students' participation in science and engineering (e.g., Burghardt & Hacker, 2004; Fortus et al., 2005; Kolodner et al., 2003; Penner, Lehrer & Schauble, 1998). These approaches strengthen scientific understanding by making science concepts consequential for students' engineering decisions and strengthen engineering by requiring students to articulate design rationales based on scientific principles. Because engineering is a relatively new addition to many precollege education frameworks, much existing research examines how engineering practices can support science learning outcomes. The relatively recent positioning of engineering as having equal footing as science in contemporary science education frameworks such as the U.S. Next Generation Science Standards (NGSS) (NGSS Lead States, 2013) motivates our design approach, which places engineering at the center and examines how science (and CT) can support it.
- **Domain-specific modeling language (DSML)**. We have created a DSML for the computational modeling activities to reduce the challenges that learners face when modeling scientific scenarios (Hasan & Biswas, 2017). The concept of DSML emerges from domain-specific language—a programming language that focuses on a particular problem domain with an appropriate level of abstraction (van Deursen, Klint, & Visser, 2000). The use of a DSML simplifies the problem by adapting the level of system abstraction to an age-appropriate level. Relative to using a general-purpose language to develop scientific models, students can place greater focus on the science concepts and system variables without having to attend to the details of model implementation.
- **Developing computational simulations for science inquiry and using them for problem-solving**. This paper integrates two strands of work on models and simulations in STEM learning. We draw from research that examines how to design computational environments that enable students to model science phenomena (e.g., diSessa, 1991; Wilensky & Rand, 2015). We also build on work that investigates how students use simulations to engage in problem-solving (e.g., de Jong & van Joolingen, 1998; Xie et al., 2018). In this paper, we merge these two perspectives—students first develop the underlying computational model driving the behavior of the simulation, then use the resulting simulation to design and test problem solutions.

A framework for integrating science, engineering, and computational modeling

Figure 1 illustrates our framework for integrating engineering, science, and CT (in the form of computational modeling). The investigation is framed around an engineering design challenge. The engineering challenge is defined so that its solution is governed by an underlying science phenomenon aligned with science learning goals for the unit. This science phenomenon must be modeled computationally so that the computational model can be used to achieve the engineering solution.

After students define and delimit the engineering problem, they explain the science phenomenon that drives the engineering problem (e.g., water runoff in the WRC). Toward this end, students investigate the phenomenon (such as using hands-on activities) and develop a conceptual model (e.g., physical representation or a drawing) that attends to the system behavior produced by the phenomenon and specifies appropriate variable relationships in mathematical terms.

After students develop a working computational model that predicts system behavior, the model output is incorporated into a simulation that supports design, testing, and optimization of engineering solutions. Students are expected to rely on a variety of testing approaches to compare multiple engineering solutions and iteratively determine solutions satisfying all design criteria.

This framework for using computational modeling to integrate science and engineering has given rise to three emergent design guidelines for designing curricular activities that leverage this framework:

- **Identify a problem and underlying phenomenon that leverages computational affordances**. To maintain authenticity in the curriculum design, the investigation should be driven by questions where computational affordances and tools are leveraged authentically as part of the science inquiry and engineering design activities. This principle promotes the CT practice of *Recognizing and defining computational problems* (K-12 CS Framework, 2016).

- **Maintain coherence across system representations.** To integrate views of the underlying science phenomenon with tools supporting engineering design, students develop multiple types of system representations, including pictorial, mathematical, narrative, and computational. These representations need to be conceptually coherent if, together, they are to help students understand the system behavior (Ainsworth, 2006). For instance, in the WRC, each system representation is explicit about the matter conservation relationship among rainfall, absorbed water, and water runoff, and what quantities constitute system inputs and outputs.
- **Ensure that a computational model can be developed using a minimal set of CT concepts and provide opportunities to learn and apply these requisite concepts.** Because our curriculum design assumes no prior programming experience, we ensure that the model can be developed using a minimal set of CT concepts. Accordingly, we provide opportunities for students to engage with those CT concepts in an unplugged (offline) context before developing computational models.

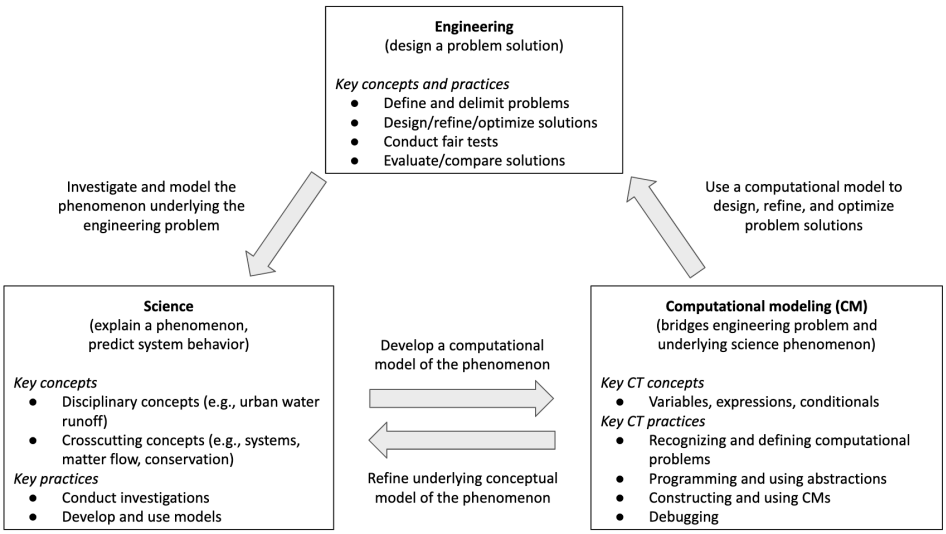

Figure 1. Relationships among engineering, science, and computational modeling in the WRC.

Curriculum unit and computational modeling environment description

The WRC is a three- to four-week unit addressing NGSS performance expectations in Earth science and engineering design. Students are presented with the problem of flooding of a schoolyard during periods of heavy rainfall. They are challenged to resurface the schoolyard in a way that minimizes the amount of surface water runoff during a storm, while also adhering to constraints on the total cost, accessibility of schoolyard, and uses of different surface materials. The science investigation involves hands-on activities contrasting absorption capabilities of different surface materials and students developing a conceptual model that expresses the amount of water runoff as the difference between the total rainfall and water absorbed by surface materials.

Students are then introduced to the CT concepts of variables, expressions, and conditionals in a series of unplugged activities. They apply these concepts to the development of a computational water runoff model and are supported in testing and debugging their code to refine their models. The computational runoff model simulates the effect of surface materials on water runoff from the schoolyard to the surrounding area and includes six materials, each having a predefined absorption limit. In the model, students must specify that when the total rainfall on any type of material exceeds the absorption limit of the material, the remainder will become runoff.

Figure 2 shows the DSML created for the computational modeling activity (left) and a correct implementation of the runoff model (right). The DSML defines a layer on top of the NetsBlox visual programming environment (Broll et al., 2018) and adapts CT concepts and practices to support the modeling of water runoff. In addition to the domain-specific constructs such as rainfall, water absorption, and runoff, the DSML blocks specify key mathematical operations (e.g., arithmetic, algebraic) and computational concepts (e.g. conditionals) to support model building.

Students' runoff model code for one part (one square) of the schoolyard is then incorporated into a simulation environment comprising 16 squares for the engineering design task (Figure 3). As part of the design task, the students need to select appropriate materials for the different parts of the schoolyard to minimize the amount of runoff. Their designs are also subject to two additional criteria: (1) not exceeding a total budget of

$750,000 and (2) having at least 6 squares built with wheelchair accessible surfaces. Because the most absorbent and accessible materials tend to have the highest cost, the students must consider trade-offs among these design variables as they refine and optimize their solutions.

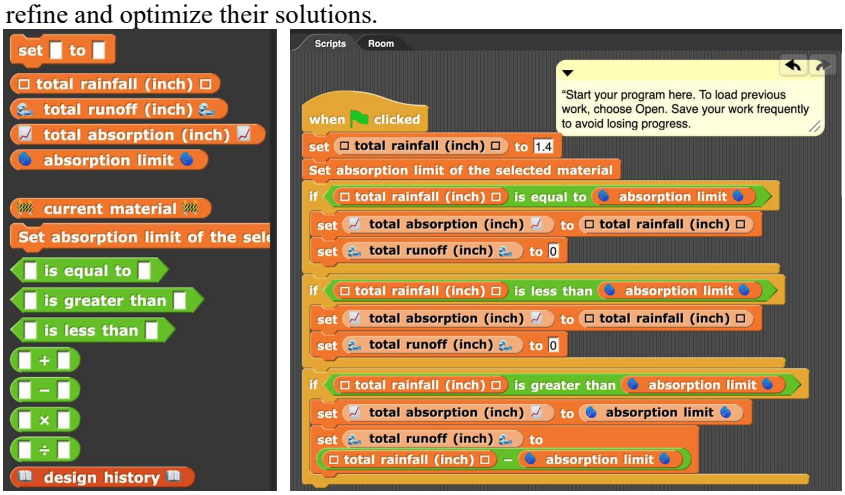

Figure 2. DSML of the runoff problem and a correct implementation of the runoff model.

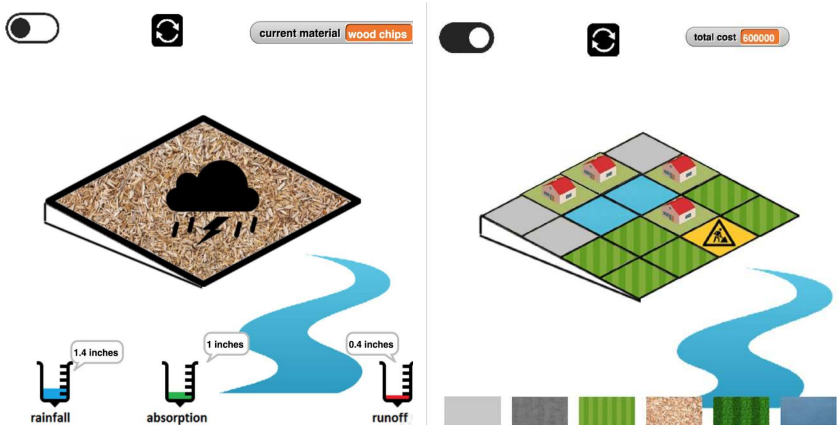

Figure 3. Screenshots of the runoff simulation environment. The left image shows the single square computational model being tested with wood chips (having a 1-inch absorption limit) using 1.4 inches of rainfall. The rain gauge icons report the amount of rainfall, absorption, and runoff. The right image shows the schoolyard simulation, where students change the design by selecting from the six available materials.

Methods

Setting and participants

We conducted a classroom study with two teachers and 99 sixth grade students (52 female, 47 male) in an academic magnet school in the southeastern U.S. serving grades 5-8. Of the total school population, 33% were from underrepresented ethnic groups in STEM and 10% were eligible for free or reduced-price lunch. The study was conducted daily for 15 school days, for 45-70 minutes per day. The students had varying degrees of prior experience in programming with a visual programming language. The two teachers participated in about four days of professional development training prior to the study, during which they completed the entire curriculum, constructed conceptual and computational runoff models, and created engineering designs of the schoolyard. The teachers were highly experienced with teaching science, but neither had any prior programming experience.

Data sources and analysis

We examined two primary types of student data: paper-based artifacts and students' actions in the computational environment. Students completed paper-based activities in class and as homework that included a range of tasks across science, engineering, and CT. The tasks captured students' understanding of the properties of different surface materials and relationships among rainfall, absorption, and surface runoff; their ability to comprehend and

debug model code; their recognition of the need to conduct fair tests of runoff designs; and their ability to compare multiple solutions relative to design criteria.

We also logged students' design and testing choices as they used the 16-square schoolyard simulation (Figure 3, right). We represented students' solution approaches as a series of points in a 3-dimensional solution space (Figures 4 and 5), with the three primary design criteria comprising the dimensions of the space: inches of water runoff, total cost, and the number of wheelchair-accessible squares contained in the design (Zhang et al., 2019). We conjectured that successful students would initially explore a wide range of combinations of surface materials, then narrow down the scope of the search in the design space to arrive at an optimized solution by using subtle refinements. To characterize the nature of students' solution approaches, we computed the Euclidean distance between points representing consecutive tests in the 3-D solution space. We computed the average distance for the first third and last third of each student's test sequence. The difference between the two, the *testing variability index,* indicates a change in solution approach from early tests to the later ones, with positive values indicating increasing variability over time.

To illustrate opportunities the WRC provided for students to integrate science, engineering, and CT, we summarize students' solution approaches and present two contrasting student vignettes. Where appropriate, we position the performance of the two students relative to the range of student performances we observed.

Findings

Students completed the WRC activities as intended. In the paper-based activities, students exhibited wide variation in their ability to make conceptual runoff models, comprehend and debug model code, and compare engineering solutions relative to design criteria. We illustrate some of this variation in the vignettes below. In the computational environment, the 99 students performed a total of 2915 tests on their schoolyard designs. The mean number of unique designs made by a student was 10.85 (SD=7.22), and a mean of 6.24 (SD=4.27) of these unique designs satisfied the given criteria for cost and wheelchair accessibility. Ninety students tested at least one satisfying design, and 29 students achieved an optimal solution representing the lowest possible water runoff for 2 inches of rainfall while simultaneously satisfying the other two criteria (0.9625 inches runoff, $750,000 cost, 6 accessible squares). These results indicate that students were generally able to create feasible engineering designs for the WRC. Additionally, the mean testing variability index for the 72 students who conducted at least 15 tests (indicating a concerted effort to test solutions) was +0.09 (SD=0.87), suggesting that students were about equally likely to vary the test variables more widely in the later tests as in the earlier ones. Below, we use two contrasting vignettes to illustrate students' learning process and performance in the WRC curriculum.

Vignette #1: Lucas

We selected Lucas (pseudonym) as the subject of the first vignette based on his high level of completion of the curricular activities and systematic testing approach using the computational environment.

Conceptual and computational modeling

From the outset, Lucas demonstrated a strong understanding of the underlying conceptual model. He made an accurate and complete pictorial model and included an accurate explanation comparing the resulting runoff from a high-absorbing and low-absorbing material. (*"I made [the models] different because grass and concrete have different absorption limits. ...[Concrete] wouldn't absorb as much rain, so the runoff would be higher.")* In addition, given a set of specific scenarios with a given amount of rainfall and an absorption limit, Lucas was able to consistently determine the expected absorption and runoff values based on a correct underlying runoff model. Only 36 of 99 students were able to determine the expected values correctly for each scenario.

In response to a series of computational model code comprehension and interpretation tasks, Lucas was also able to correctly predict the absorption and runoff output for incompletely and/or incorrectly coded computational runoff models and describe how to elaborate or debug the code. ("*One error is that she said if absorption limit is less than total rainfall. That should be switched. Also, it should set total absorption to total rainfall, not the other way around.*") This performance illustrates the extent to which Lucas was able to integrate science modeling and CT by applying his understanding of the conceptual runoff model to code comprehension and debugging tasks.

Engineering design

In the paper-and-pencil engineering tasks, Lucas demonstrated a strong understanding of the central engineering concepts of design trade-offs and fair tests. Asked to compare two hypothetical schoolyard designs, Lucas was among 10 students who explicitly discussed the trade-off between runoff performance and cost. ("*[Onir's design] would be the better design because it costs less money. Although [Onir's] has more runoff, Pelayo's model costs*

more money than the school could handle.") Lucas was also among the 20 students who identified a flaw in a given design test ("*She can't tell which design is better because she did not have the same rainfall inputs for both of them. So, the outputs would be different*") and correctly justified a new test based on the need for fair tests.

In using the simulation to develop, test, and optimize designs, Lucas was one of the 29 students who achieved the optimal design solution. Consistent with our conjecture, Lucas's solution approach using the simulation (Figure 4) exhibited two clear phases: broad exploration characterized by highly contrasting designs that reach the boundary values of the solution space, followed by a series of smaller changes representing apparent attempts to optimize the solution iteratively. Lucas's testing variability index of -1.50 was the seventh lowest achieved by any student in the study, indicating a decrease in variability over time. Using the two-phase approach, Lucas was able to test both low-cost/high-runoff (tests 5-7) and high-cost/low-runoff (test 11) solutions that exemplify a central trade-off students must make in this design problem. Lucas proceeded to the optimization phase, where he returned to the optimal solution six different times (tests 17, 33, 35, 39, 42, and 46) and avoided testing clearly infeasible solutions. This approach was informed by an underlying understanding of the relationship among surface materials, absorption limits, and resulting runoff, as Lucas was able to narrow his problem space search in a way that focused on solutions exhibiting good runoff performance.

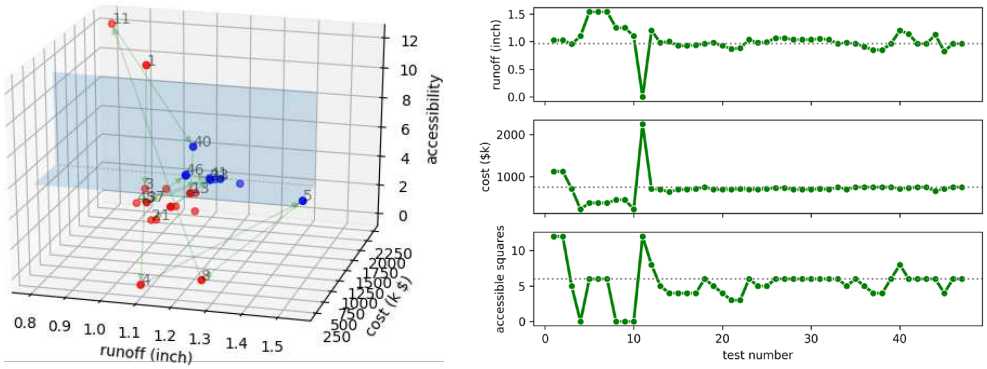

Figure 4. Lucas's solution testing sequence. Blue markers represent designs satisfying all criteria. The dashed lines on the 2-D plots represent the optimal runoff performance for a satisfying solution (0.9625 inches), and the cost ($750,000) and accessibility (6 squares) criteria. The plots reveal two distinct solution approaches: exploration (tests 1-12) and optimization (tests 13-47).

Vignette #2: Adam

Adam (pseudonym) was less successful with the paper-based tasks and used a less systematic testing approach than Lucas. Adam struggled with some of the initial conceptual modeling tasks; for example, Adam indicated 2 inches of water absorption in a model where there was only 1 inch of rainfall. He was also unable to correctly predict the output of given runoff model code. In the computational environment, Adam did not focus his solution search in a way that is consistent with either an understanding of the underlying runoff model or the computational affordances of the environment for solution refinement (Figure 5). Adam's testing variability index of +1.18 was the 11th highest of all students in the study, indicating an increase in variability over time. Part of this variability increase was a result of Adam frequently repeating tests early in the search. However, Adam's late-stage tests clearly exhibit drastically diverging designs having either poor runoff performance (tests 35, 40, 43, and 44) or low accessibility (test 38, 39, 41, and 42). Though Adam was able to achieve several satisfying solutions, his testing approach did not iteratively improve on any of them, and at no time during the search did Adam reach the optimal solution.

Discussion and conclusions

Our analysis illustrates opportunities students have to use both "science" and "engineering" experimentation approaches (e.g., Schauble, Klopfer, & Raghavan, 1991). Science approaches aim to characterize variable relationships by systematically establishing the effect of each potentially relevant variable, while engineering approaches aim to achieve specific outcomes by comparing contrastive cases and refining promising solutions. The WRC presents students with the "semi-specific" goal to *minimize* runoff (rather than achieve a specific threshold). Solution optimization can, therefore, benefit from science-oriented approaches that systematically examine individual design variables, rather than compelling students to stop testing after achieving a specific goal. We also highlight the important role of prior knowledge in how learners test problem solutions (e.g., Klahr & Dunbar, 1988). In the WRC, the conceptual runoff model (relating materials to their absorption capacity and

consequently to runoff performance) enables students to rapidly pare down the problem space and use an optimization approach reflecting the relationships among design variables such as runoff performance and cost. We conjecture that providing students like Adam with adaptive, just-in-time guidance could help them conduct tests more informatively. For example, such guidance could help students articulate trade-offs between design variables and identify promising solutions to optimize in subsequent tests.

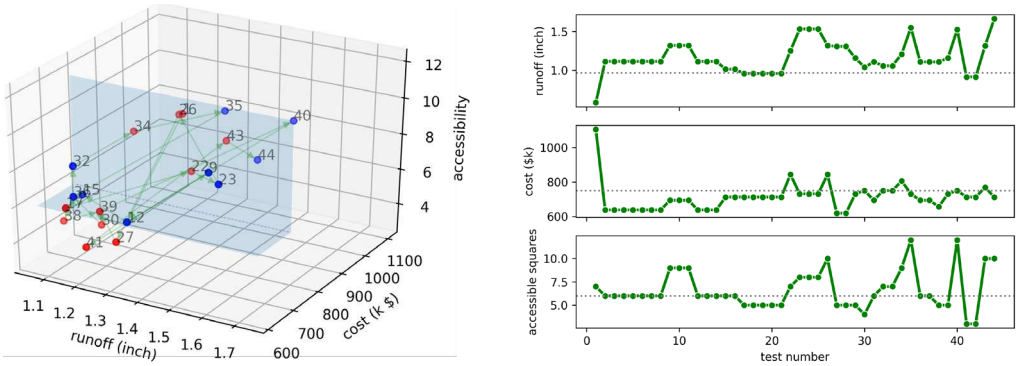

Figure 5. Plots representing Adam's solution testing sequence, where variation between consecutive tests increases over time.

We also note the affordances of CT for supporting engineering design processes. Enabling students to develop an underlying scientific computational model before using it to solve a related engineering problem contributes to the authenticity of the activity, provides students with greater agency, and appropriately demonstrates the role of CT in professional science and engineering. This approach, rather than presenting the simulation as a "black box," allows students to determine for themselves "what's under the hood," consequently reinforcing the scientific concepts driving the simulation and helping students make the scientific rationale behind design decisions explicit. Our approach also aligns with frameworks recommending *constructive* learning activities over *active* ones (e.g., Chi & Wylie, 2014). In the WRC, because students have constructed the underlying model driving the schoolyard simulation, students' subsequent use of the simulation to test solutions becomes a meaningful extension of a prior constructive activity, rather than merely an active one.

We present a design framework for using computational modeling to coherently integrate science and engineering curricular activities and provide guidelines for instructional designers to apply our framework to new activity designs. The framework uses computational modeling perspectives to merge two salient research traditions in STEM learning: developing computational models of science phenomena and using computational simulations to solve problems. The framework responds to recent calls to integrate learning opportunities across STEM and computer science driven by the vanishing boundaries between disciplines in professional practice. Future work will entail analysis of pre/posttest assessments that measure student learning across science, engineering, and CT; analyses of students' approaches to developing computational models; and examining factors that promote successful classroom implementation of integrated STEM curriculum materials.

References

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

Basu, S., Kinnebrew, J. S., Dickes, A., Farris, A., Sengupta, P., Winger, J., & Biswas, G. (2012). A science learning environment using a computational thinking approach. In *20th International Conference on Computers in Education, ICCE 2012*.

Basu, S., Biswas, G., & Kinnebrew, J. S. (2017). Learner modeling for adaptive scaffolding in a computational thinking-based science learning environment. *User Modeling and User-Adapted Interaction*, *27*(1), 5-53.

Broll, B. et al. (2018). A visual programming environment for introducing distributed computing to secondary education. *Journal of Parallel and Distributed Computing*, *118*, 189-200.

Burghardt, M. D., & Hacker, M. (2004). Informed design: A contemporary approach to design pedagogy as the core process in technology. *Technology teacher*, *64*(1), 6-8.

Chi, M. T., & Wylie, R. (2014). The ICAP framework: Linking cognitive engagement to active learning outcomes. *Educational psychologist*, *49*(4), 219-243.

Chiu, J., McElhaney, K. W., Zhang, N., Biswas, G., Fried, R., Basu, S., & Alozie, N. (2019, April). A principled approach to NGSS-aligned curriculum development integrating science, engineering, and computation: a pilot study. Paper presented at the *NARST Annual International Conference*, Baltimore, MD.

de Jong, T., & van Joolingen, W. R. (1998). Scientific discovery learning with computer simulations of conceptual domains. *Review of educational research*, *68*(2), 179-201.

diSessa, A. A. (1991). An overview of Boxer. *Journal of Mathematical Behavior*, *10*(1), 3-15.

Fortus, D., Krajcik, J., Dershimer, R. C., Marx, R. W., & Mamlok-Naaman, R. (2005). Design-based science and real-world problem-solving. *International Journal of Science Education*, *27*(7), 855-879.

Harris, C. J., Krajcik, J., Pellegrino, J. & DeBarger, A. H. (2019). Designing knowledge-in-use assessments to promote deeper learning. *Educational Measurement: Issues and Practice.* 38 (2), 53-67.

Hasan, A., & Biswas, G. (2017). Domain specific modeling language design to support synergistic learning of STEM and computational thinking. In *S. C. Kong, J. Sheldon, & K. Y. Li (Eds.). Conference Proceedings of the International Conference on Computational Thinking Education 2017* (pp. 28-33). The Education University of Hong Kong: Hong Kong.

Klahr, D., & Dunbar, K. (1988). Dual space search during scientific reasoning. *Cognitive science*, *12*(1), 1-48.

Kolodner, J. L., et al. (2003). Problem-based learning meets case-based reasoning in the middle-school science classroom: Putting Learning by Design™ into practice. *The Journal of the Learning Sciences*, *12*(4), 495-547.

Krajcik, J., McNeill, K. L., & Reiser, B. J. (2008). Learning-goals-driven design model: Developing curriculum materials that align with national standards and incorporate project-based pedagogy. *Science Education*, *92*(1), 1-32.

K–12 Computer Science Framework. (2016). Retrieved from http://www.k12cs.org.

Lehrer, R., & Schauble, L. (2006). *Cultivating model-based reasoning in science education.* Cambridge University Press.

Linn, M. C., & Eylon, B-S. (2011). Science learning and instruction: Taking advantage of technology to promote knowledge integration. New York, NY: Routledge.

Mislevy, R. J., & Haertel, G. D. (2006). Implications of evidence-centered design for educational testing. *Educational Measurement: Issues and Practice*, *25*(4), 6-20.

National Research Council. (2011). *Report of a workshop on the pedagogical aspects of computational thinking*. National Academies Press.

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

Penner, D. E., Lehrer, R., & Schauble, L. (1998). From physical models to biomechanics: A design-based modeling approach. *Journal of the Learning Sciences*, *7*(3-4), 429-449.

Schauble, L., Klopfer, L. E., & Raghavan, K. (1991). Students' transition from an engineering model to a science model of experimentation. *Journal of research in science teaching*, *28*(9), 859-882.

van Deursen, A., Klint, P., & Visser, J. (2000). Domain-specific languages: An annotated bibliography. *ACM Sigplan Notices*, 35(6), 26-36.

Wilensky, U., & Rand, W. (2015). *An introduction to agent-based modeling: modeling natural, social, and engineered complex systems with NetLogo*. MIT Press.

Wilkerson-Jerde, M., Wagh, A., & Wilensky, U. (2015). Balancing curricular and pedagogical needs in computational construction kits: Lessons from the DeltaTick project. *Science Education*, *99*(3), 465-499.

Wing, J. M. (2008). Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, *366*(1881), 3717-3725.

Xie, C., Schimpf, C., Chao, J., Nourian, S., & Massicotte, J. (2018). Learning and teaching engineering design through modeling and simulation on a CAD platform. *Computer Applications in Engineering Education*, *26*(4), 824-840.

Zhang N., Biswas G., Chiu J.L., & McElhaney K.W. (2019). Analyzing Students' Design Solutions in an NGSS-Aligned Earth Sciences Curriculum. In S. Isotani, E. Millán, A. Ogan, P. Hastings, B. McLaren, R. Luckin (Eds). *Proceedings of Artificial Intelligence in Education. AIED 2019.* Springer: Cham, Switzerland.

Acknowledgments

We gratefully acknowledge the support of National Science Foundation award DRL-1742195. We also appreciate the contributions of other research team members 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.