




Studying the Interactions Between Science, Engineering, and Computational Thinking in a Learning-by-Modeling Environment

Ningyu Zhang¹ , Gautam Biswas¹, Kevin W. McElhaney², Satabdi Basu³, Elizabeth McBride³, and Jennifer L. Chiu⁴

¹ Department of EECS, Institute for Software Integrated Systems, Vanderbilt University, 1025 16th Avenue South, Nashville, TN 37212, USA
{ningyu.zhang,gautam.biswas}@vanderbilt.edu

² Learning Sciences Research, Digital Promise, San Mateo, USA
kmcElhaney@digitalpromise.org

³ Center for Education Research and Innovation, SRI International, Menlo Park, USA
{satabdi.basu,beth.mcbride}@sri.com

⁴ Department of Curriculum, Instruction and Special Education, University of Virginia, Charlottesville, USA
jlchiu@virginia.edu

Abstract. Computational Thinking (CT) can play a central role in fostering students' integrated learning of science and engineering. We adopt this framework to design and develop the Water Runoff Challenge (WRC) curriculum for lower middle school students in the USA. This paper presents (1) the WRC curriculum implemented in an integrated computational modeling and engineering design environment and (2) formative and summative assessments used to evaluate learner's science, engineering, and CT skills as they progress through the curriculum. We derived a series of performance measures associated with student learning from system log data and the assessments. By applying Path Analysis we found significant relations between measures of science, engineering, and CT learning, indicating that they are mutually supportive of learning across these disciplines.

Keywords: Science and engineering · Computational modeling · Log analysis · Regression methods

1 Introduction

The Next Generation Science Standards (NGSS) call for the inclusion of engineering design activities in K-12 science classrooms and propose that science investigation and engineering design be closely integrated into the curriculum [19, 22]. In addition, computational modeling and analysis have become

a key component of scientific study [8]. We adopt an integrated approach to developing science and engineering curricula, bringing in computational thinking (CT) concepts through computational modeling activities to develop the Water Runoff Challenge (WRC) for fifth and sixth grade students [4,34].

This paper discusses the WRC curriculum, the learning environment that supports the computational modeling and engineering design activities, and the formative and summative assessments developed for evaluating student learning. We discuss the results of a study with 99 sixth-grade students. The intervention produced significant learning gains in science, engineering, and CT with moderate to large effect sizes. Given these results, we applied Path Analysis [1,33] to model the relationships between measures of student learning in science, engineering, and CT, and interpreted their relative importance. In more detail, we derived a range of measures from logs of student activities and their assessment scores to investigate (1) the relations between students' behavior and performance variables in the computational modeling and engineering design activities and (2) which of these variables contribute to the learning outcomes. Path analysis also informs us of the importance and significance of pairwise relations.

1.1 Background and Related Work

The majority of people in the U.S. are introduced to science and engineering in middle and high schools, and the experiences in these formative years shape their interest in pursuing science and engineering careers [15,25]. However, engineering had not traditionally been part of the core K-12 curriculum, instead often being offered as an elective or after-school course, where students primarily work on design projects with little discussion of the science that supports the design and implementation [6].

Recently, the “growing inclusion of engineering design in K-12 classrooms” presents students with opportunities to construct an understanding of the natural and designed world [19, p. vii]. It has been proposed that *science investigation*, which includes students' investigating scientific phenomena and *engineering design*, i.e., applying the learned knowledge to design solutions to challenges of interest should be more central to the K-12 curricula [19].

Modeling is a key practice in science and an essential mechanism to support effective engineering design [20,24,28]. A model is defined as an abstract and simplified representation of a scientific phenomenon built around the important features that explain and predict the phenomenon [9,28]. *Computational modeling* has become integral to STEM learning and practice [21,30]. Computational modeling activities can support the learning of science and engineering in virtual environments by (1) enabling learners to manipulate variables on unobservable phenomena and (2) improving the efficiency and reducing unanticipated consequences of experimental studies [7]. In other words, learners have more opportunities to conduct systematic investigations and gather more information as compared to conducting observations in a physical environment [7]. For example, chemical reactions (invisible) and geological changes (long-term) are easier to study by simulating computational models than trying to conduct physical or

observational studies. Students' engagement with computational modeling activities provides instructional benefits of improved domain knowledge and problem-solving skills [2, 29, 31].

2 The Learning Environment in the WRC Curriculum

Our research is motivated by the trends towards integrated learning of science and engineering. Furthermore, we introduce CT and computational modeling activities as a platform for integrated engineering and science learning [30, 32]. *Exploration* with models involves manipulating parameters to study the model's behaviors. On the other hand, model exploration often does not require invoking the complex cognitive processes required to build models, which includes scoping the model, developing algorithms to represent model behaviors, computing numerical outcomes, interpreting results, and validating solutions; neither do students have to fully understand the nuances of the modeling language employed [18, 29]. In our *previous* work, students used a pre-built computational model for their engineering task to develop and test playground designs to mitigate flooding problems in a school [4, 34]. In the present work, we introduce computational model building activities into the WRC curriculum unit. Students used the runoff computational models that they developed themselves to design a schoolyard that reduced runoff and its associated environmental impact.

Our previous runoff model was *dynamic*; the model representations needed to capture the behavior of a system over time [29]. An *agent-based approach* to modeling [5] makes the model modular and facilitates decomposition into its constituent parts. For compatibility with middle school math proficiency, the systems dynamics model was simplified to a discrete-time algebraic form [29]. This simplified representation computes the amount of rainfall, absorption, and runoff with three simultaneous equations. To make this form of modeling representation explicit and linked to the science concepts, we have created a domain-specific modeling language (DSML) to support students' computational modeling activities [10]. DSMLs specify modeling constructs at a level of abstraction that is compatible with the students' ability to build and analyze the model.

Figure 1 shows the DSML blocks created for the computational modeling activity on the left, and a correct implementation of the runoff model using these blocks on the right. The DSML, created in the NetsBlox visual programming environment [3], incorporates CT concepts, such as control structures along with the primary domain concepts to support the modeling of the water runoff processes: (1) the amount of rainfall (2) absorption of water by different surface materials, and (3) runoff. In addition, the DSML specifies key arithmetic and algebraic mathematical operations to support model-building. Using the DSML blocks, students create a rule-based computational model, which is a simplification of the system dynamics model. The runoff for a specific material is computed as the difference in the amount of rainfall and the amount of water that is absorbed by the surface material (see the example implementation in Fig. 1).

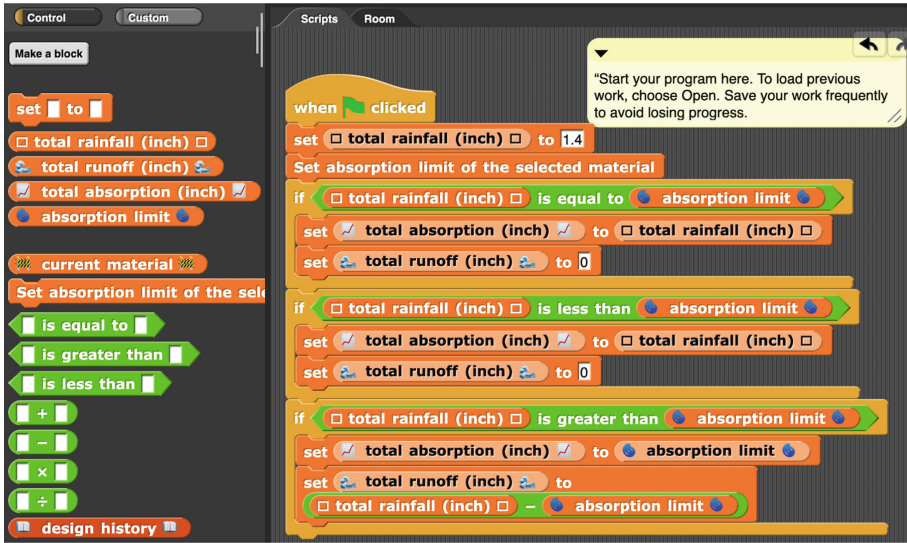


Fig. 1. DSML of the runoff problem (left) and a correct implementation of the runoff model using the DSML (right).

Students build schoolyards models for their engineering design tasks. They do this using a visual interface to populate individual squares with different surface materials (Fig. 2). The computational models that the students develop are used to calculate the total absorption and total runoff given a total amount of rainfall that the student specifies. The students can build and test multiple schoolyard designs using different combinations of materials. Their overall goal is to (1) minimize runoff, (2) remain under budget; and (3) ensure that sufficient squares in the schoolyard have accessible surface materials to meet wheelchair needs. Students need to generate multiple designs using a search process to find the *optimal* design that meets all of the constraints, i.e., minimize runoff, while meeting the cost and accessibility constraints. This design task is challenging for young learners. Typically, the more absorbent and accessible materials also tend to have high costs, so students need to analyze the trade-offs between cost, absorption, and accessibility in searching for optimal design solutions. A non-systematic *trial-and-error* approach may overwhelm a student's search. Figure 2 (right) depicts the engineering design interface. The current solution is incomplete, and students can assign any of the six available materials to the unassigned yellow square.

3 Methods

We conducted a classroom study with 99 sixth-grade middle school students in the U.S. using the WRC curriculum. All participating students had varying levels of prior programming experience with block-structured programming

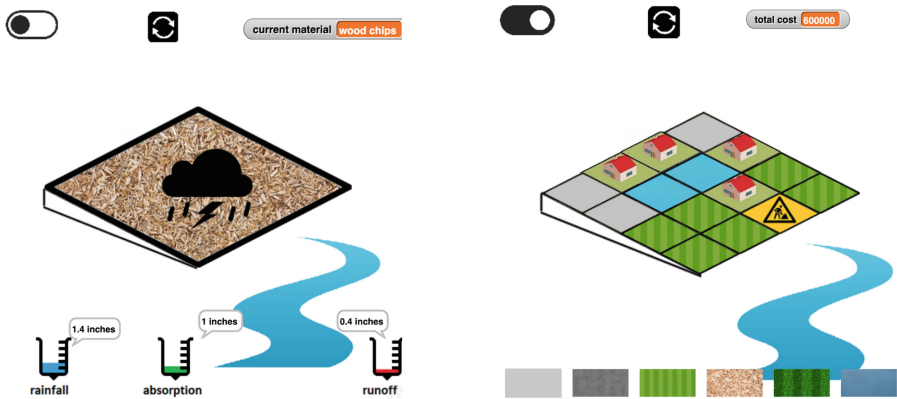


Fig. 2. The runoff simulation (left) and material selection (right) interface (Color figure online)

with Scratch [26] in their programming class. The study was led by two experienced science teachers who received four days of training before the study. Three researchers provided additional support but mostly acted as observers during the study. Students worked for 45 min per day, three days a week during their regular science classes, and 75 min, twice a week with additional personalized-learning time. The WRC curriculum was covered in 15 school days, with identical pre-post tests administered in two additional 45-min classes.

The WRC unit also includes (1) hands-on activities in which students conduct physical investigations on the absorption of different surface materials; (2) conceptual modeling of the runoff system as a pictorial representation; and (3) presenting their methods and final engineering designs. This paper analyzes the NGSS-aligned science and engineering + CT pre-post assessments and the data collected on days 8–13. This includes (1) formative assessments administered as homework that covered science, engineering, and CT topics; and (2) system logs of students' model-building and engineering design activities.

Assessments and Grading. Our science and engineering summative assessments align with a number of NGSS Performance Expectations (PEs) [16,17]. The CT assessments are derived from the concepts and practices that students perform as part of their science modeling activities. The rubrics used for coding and scoring these assessments were updated from our previous work [16]. Two researchers received 5 h of training on the rubrics, during which 5% of the test submissions were randomly selected and graded together to establish initial grading consistency. Another 20% test submissions were then graded by the two researchers independently 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. We also designed *formative* assessment tasks that mirror the

curricular tasks students worked on in the WRC. These tasks measured students' understanding of (1) the water conservation relations, (2) the relative effect of different surface materials on runoff, (3) the ability to compute water runoff and absorption under different circumstances, (4) the ability to debug incomplete or incorrect model code, and (5) the method to compare different design solutions considering trade-offs. We used students' responses to 14 items from 6 formative assessment tasks in this work.

Log Analysis. The learning environment logs individual students' actions during their computational model-building and engineering design activities. We calculated three behavioral measures from students' **computational modeling** activities: (1) the total number of add, remove, connect, or disconnect blocks actions, (2) the number of run the model actions to test the computational model, and (3) the median number of edit actions between tests (because students often perform a series of edits without testing or a series of testing without editing the model, the median number is a more robust measure given the skewness in the data). In addition to deriving behavior measures, we defined a computational model score for the student-generated models. A correct computational model scored 6 points (1 point for each correctly implemented function that calculates and assigns values to an output variable. There were two variables each in the three rules, see Fig. 1 for reference). To allow students to conduct meaningful design activities, the researchers made an effort to ensure that all students' had correct computational models before they started the design activity. Common errors were discussed with the whole class, and the students were given a chance to correct their models. The model scores reported in this work were calculated *before* the correction feedback was provided to the students.

Our measurements of students' *engineering design* quality and their learning behaviors have been discussed in our previous work [34]. The two quality measurements used are: (1) the number of satisfying designs and (2) the smallest runoff value from all of the satisfying designs created by a student. The two behavior measurements used are: (1) the number of tests conducted to evaluate designs; and (2) the total standardized Euclidean distance between a student's m consecutive tested designs, i.e., $\sum_{i=1}^{m-1} \|(V_{i+1} - V_i)^2\|$ where $V = \langle runoff_z, cost_z, accessibility_z \rangle$. The subscript z indicates the standardized value of runoff, cost, and accessibility of a design. The total standardized Euclidean distance and the number of tested designs indicate the extent to which a learner explored the engineering design *experiment space* [12].

Path Analysis. Traditional regression methods assume that (1) only direct associations exist between dependent and independent variables and (2) errors in the dependent variable are uncorrelated with the independent variable [1, 33]. When applied to intrinsically related variables, where indirect variables play a mitigating role, multi-regression or correlation analysis do not provide optimal model estimates [23]. Path Analysis addresses these problems. It can be seen as a variation of Structural Equation Modeling [13] without the latent variables. In this work, we use Path Analysis to study the effects and the relative importance

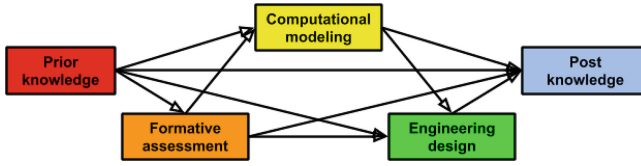


Fig. 3. Hypothesized path model.

of effects among the measured performance and behavior values. We hypothesize that students’ prior knowledge and formative assessment scores influence their subsequent learning behaviors, computational model building and engineering design performance, and post-test scores in the WRC curriculum. This is represented by the causal path model shown in Fig. 3. Each arrow in the diagram indicates a direct effect on the endogenous variable from the exogenous variable.

4 Results and Discussions

4.1 Learning Performances and Behaviors

Students’ pre-post test scores were compared to determine their learning gains in science, engineering, and CT. To check the normality of the scores, we first measured the skewness (z -value = -0.811 , p -value = 0.417) and kurtosis (z -value = -0.567 , p -value = 0.571) of the score distributions and confirmed that they were close to a normal distribution. Therefore, we used the paired t -test to evaluate the statistical significance of the pre-post score differences. Table 1 shows that all differences are statistically significant with moderate (≥ 0.5) to large (≥ 0.8) effect sizes.

Table 1. Learning gains (N = 99)

	Total points	Pre-score (<i>stdev</i>)	Post-score (<i>stdev</i>)	<i>p</i> -value	Cohen’s <i>d</i>
Science	7	4.56 (1.03)	5.13 (1.04)	<0.001	0.54
Engineering	16	8.73 (2.62)	10.50 (2.67)	<0.0001	0.67
CT	13	6.23 (2.60)	8.41 (2.69)	<0.0001	0.83
Overall	36	19.52 (4.47)	24.03 (4.39)	<0.0001	1.02

Formative Assessment. The average score of the integrated science, engineering, and CT formative assessment was 19.05 points ($stdev = 4.57$) out of maximum possible scores of 31 points. This result, along with the pre-post test gains, indicates that the students were learning the domain content, CT concepts and skills, and engineering design practices through the intervention.

Computational Modeling. Students showed a large variation in their computational model-building behaviors. On average, they made 167 edits ($stdev = 77$)

to build their computational models, and they performed 43 tests ($stdev = 47$) on them. The average of the median number of edits between tests was 1.11, indicating the student mostly made edits in small chunks between successive model tests. The average computational model score was 4.67 ($stdev = 1.85$), and 59% of the students created a correct computational model before the answer was disclosed in class. The model component with the least number of correct implementations ($n = 67$) was “set total runoff to ($total\ rainfall - absorption\ limit$)” when “total rainfall is greater than absorption limit” (c.f. Fig. 1).

Engineering Design. The students performed an average of 29.4 tests ($stdev = 22.2$) on their schoolyard designs. The average total standardized Euclidean distance was 18.6 ($stdev = 19.0$). The average number of unique designs that satisfied the criteria for cost and accessibility was 6.3 ($stdev = 4.2$). Ninety students created and tested at least 1 satisfying design, and the average amount of runoff for the satisfying design solutions, with 2 inch of rainfall, was 1.23 inch ($stdev = 0.94$). The global minimal runoff of all satisfying designs was 0.96 inch, and 29 students got at this optimal solution. These results show that most students created feasible design solutions.

4.2 Path Analysis

We created a path diagram of the measured variables using the IBM[®] SPSS[®] Amos 26 software. We modeled a total of 47 direct effects from the 15 variables in the path diagram. As a pre-analysis suggested by [27], we evaluated the assumptions of multivariate normality and then removed four outliers from subsequent analyses, leaving a sample size of 95 for the Path Analysis. 1000 bootstrap samples were generated to estimate the standard errors and calculate the confidence intervals at the 95% level. The standard errors and their critical ratios were later used to evaluate the statistical significance of the modeled causal effects while reducing the variance in the observed variables.

We also calculated model-fitting statistics of the path model as compared to the saturated model [27]: $\chi^2 = 40.89$ ($DF = 54$, p -value = 0.91); the goodness of fit (GFI) was 0.95 (≥ 0.95 threshold); the comparative fit index (CFI) was 0.99 (> 0.9 threshold); and the root mean square error of approximation (RMSEA) was 0.01 (< 0.06 threshold). These statistics indicate that the path model derived fitted the measurements well. All of the hypothesized paths in Fig. 3 were confirmed as direct or indirect effects. Figure 4 shows the statistically significant causal paths that are large ($\beta > 0.2$).

Computational Modeling. The students’ learning behaviors and performance in the computational modeling activity (yellow boxes in Fig. 4) were directly affected by variables in the same category and the formative assessment score. The CT pre-test score also indirectly related to the *comp.model.score* and *comp.edits* (via *formative*, *comp.test*, and *edit.btw.tests*) with total β ’s of 0.28 and 0.28, respectively (indirect effects are not shown in Fig. 4). As one of the

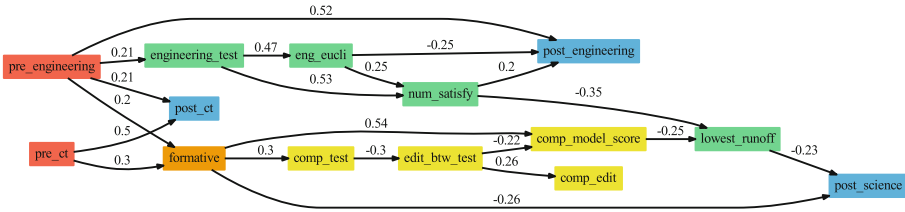


Fig. 4. Discovered causal paths with statistically significant direct effects. (Color figure online)

main learning outcomes, the students’ *comp_model_score* was also significantly affected by the median number of model edits between tests (*edit_btwt_test*), indicating students who edited their model in *small* chunks between tests did better in the computational model-building task. Similar results of smaller edit chunks being associated with better models have also been reported by [2]. The engineering pre-test score (*pre_eng*) also had a statistically significant but small (total $\beta = 0.12$) indirect effect on *comp_model_score* through *formative*.

Engineering Design. The number of unique satisfying designs (*num_satisfy*) and the lowest amount of runoff of satisfying designs (*lowest_runoff*) were the two variables evaluating the quality of students’ designs. For *num_satisfy*, the strongest direct effects came from the number of tests on the designs (*engineering_test*, $\beta = 0.53$) and the total standardized Euclidean distance between the tested designs (*eng_euclid*, $\beta = 0.25$). The *lowest_runoff* was most strongly affected by *num_satisfy* ($\beta = -0.35$) and *comp_model_score* ($\beta = -0.25$).

These results align with our previous findings with a group of fifth-grade students in another school that students who explored a larger portion of the problem space were more likely to generate better engineering design solutions [34]. It also matched the *scientific discovery as dual search* theory [12] that successful learners connect the *hypothesis space* and the *experiment space* by making inferences with data drawn from their investigations. More importantly, these results suggest a strong connection between computational modeling (*comp_model_score*) and engineering design (*lowest_runoff*) with a total standardized effect of -0.32 ($\beta = -0.25$, total indirect effect is -0.07). The negative value indicates that students making better computational models on their own generated better design solutions, even though all students were shown the correct implementation of the computational model before the engineering design activity. It also indicates the benefits of having students develop their own computational model to use for designing and testing, relative to providing students with a model that has been developed by experts.

Post-test Scores. The **science** post-test scores (*post_sci*) were significantly influenced by *lowest_runoff* ($\beta = -0.23$), *num_satisfy* (indirectly, total β ’s = 0.08), *engineering_test* (indirectly, total $\beta = 0.08$), and *comp_model_score* (indirectly, total $\beta = 0.04$). The **engineering** post-test scores were mostly affected

by *pre_eng* ($\beta = 0.52$), *eng_euclid* ($\beta = -0.25$), and *num_satisfy* (indirectly, total β 's = 0.20). The effect from *num_satisfy* indicates that students' success in solving the engineering design problem by searching for the optimal combinations of surface materials on the schoolyard reflected better learning outcomes. As of the **CT** post-test score, it was only significantly affected by the related pre-test scores. The variable *comp_model_score* had a relatively large total effect of 0.14 on *post_ct* yet the effect was not statistically significant.

These overall positive results suggest that the students' success with the engineering design activities can be linked to their science and engineering proficiencies, providing evidence for the benefit of integrating engineering with science learning [19]. In addition, the effect of engineering activities on the summative assessments suggested that the design goals of the WRC curriculum were achieved, and students' high learning gains (Cohen's $d = 1.02$) illustrated the benefits of integrating instruction across engineering and science.

Future Work. In the present work, we identified the connections between computational modeling, engineering design, and the learning outcomes as *effects* on the causal paths. Such connections might not be discovered by only examining the associations between the variables using model-less correlation methods [23]. For example, the correlation coefficient (Spearman's ρ) between *comp_model_score* and *lowest_runoff* was -0.11 ($p = 0.28$). This suggests that Path Analysis is an effective technique to study the relationship between related variables, such as the measures derived from the WRC.

This work can be further advanced by employing more sophisticated measures. For example, we used a simple heuristic to measure the computational modeling performance. We plan to (1) implement more sophisticated methods to study the structure of the students' models (e.g., abstract syntax trees (ASTs) [14]) and (2) include machine learning methods (e.g., sequence mining [11]) to analyze and understand their learning processes and learning strategies. These measures will help us design online feedback in the system to support student learning.

5 Conclusions

The Water Runoff Challenge is one of the first examples of NGSS-aligned curricula that support the interdisciplinary learning of science, engineering, and CT. In the present work, the curriculum is enhanced by enabling computational modeling activities for students to develop and practice CT instead of performing engineering design with a pre-built model. Results from our classroom study demonstrated the instructional benefits of using the WRC and provided empirical evidence to support the integration of engineering activities with science learning and computational model building, especially in early K-12 settings.

Our studies point to ways that using computational modeling to integrate science and engineering can merge insights from two learning research traditions: developing computational artifacts and engaging in simulation-based problem-solving. Specifically, our analysis suggests potential benefits of guiding students'

development of a computational scientific model prior to using the model to solve a related engineering problem. Further research is needed to better understand the learning processes that produce such benefits and identify instructional design features that best take advantage of them.

Acknowledgment. This material is based upon work supported by the National Science Foundation under Grant No. DRL-1742195. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

1. Ahn, J.: Beyond single equation regression analysis: path analysis and multi-stage regression analysis. *Am. J. Pharm. Educ.* **66**(1), 37–41 (2002)
2. Basu, S., Biswas, G., Kinnebrew, J.S.: Learner modeling for adaptive scaffolding in a computational thinking-based science learning environment. *User Model. User Adap. Inter.* **27**(1), 5–53 (2017)
3. Broll, B., et al.: A visual programming environment for introducing distributed computing to secondary education. *J. Parallel Distrib. Comput.* **118**, 189–200 (2018)
4. Chiu, J., et al.: A principled approach to NGSS-aligned curriculum development integrating science, engineering, and computation: a pilot study. Paper Presented at the 2019 NARST Annual International Conference (2019)
5. Collins, A., Ferguson, W.: Epistemic forms and epistemic games: structures and strategies to guide inquiry. *Educ. Psychol.* **28**(1), 25–42 (1993)
6. Cunningham, C.M., Knight, M.T., Carlsen, W.S., Kelly, G.: Integrating engineering in middle and high school classrooms. *Int. J. Eng. Educ.* **23**(1), 3 (2007)
7. de Jong, T., Linn, M.C., Zacharia, Z.C.: Physical and virtual laboratories in science and engineering education, April 2013. <https://doi.org/10.1126/science.1230579>. <http://www.sciencemag.org/cgi/doi/10.1126/science.1230579>
8. Denning, P.J.: The profession of IT beyond computational thinking. *Commun. ACM* **52**(6), 28–30 (2009)
9. Harrison, A.G., Treagust, D.F.: A typology of school science models. *Int. J. Sci. Educ.* **22**(9), 1011–1026 (2000)
10. Hutchins, N., Biswas, G., Zhang, N., Snyder, C., Lédeczi, Á., Maróti, M.: Domain-specific modeling languages in computer-based learning environments: a systematic approach to scaffold science learning through computational modeling. *Int. J. Artif. Intell. Educ.* (submitted, under review)
11. Kinnebrew, J.S., Loretz, K.M., Biswas, G.: A contextualized, differential sequence mining method to derive students' learning behavior patterns. *JEDM J. Educ. Data Min.* **5**(1), 190–219 (2013)
12. Klahr, D., Dunbar, K.: Dual space search during scientific reasoning. *Cogn. Sci.* **12**(1), 1–48 (1988)
13. Kline, R.B.: *Principles and Practice of Structural Equation Modeling*. Guilford Publications, New York (2015)
14. Lazar, T., Možina, M., Bratko, I.: Automatic extraction of AST patterns for debugging student programs. In: André, E., Baker, R., Hu, X., Rodrigo, M.M.T., du Boulay, B. (eds.) *AIED 2017. LNCS (LNAI)*, vol. 10331, pp. 162–174. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-61425-0_14

15. Maltese, A.V., Tai, R.H.: Eyeballs in the fridge: sources of early interest in science. *Int. J. Sci. Educ.* **32**(5), 669–685 (2010)
16. McElhaney, K.W., Basu, S., Wetzels, T., Boyce, J.: Three-dimensional assessment of NGSS upper elementary engineering design performance expectations. Paper Presented at the 2019 NARST Annual International Conference (2019)
17. McElhaney, K.W., Zhang, N., Basu, S., McBride, E., Biswas, G., Chiu, J.: Using computational modeling to integrate science and engineering curricular activities. In: *Proceedings of the International Conference of Learning Sciences, Nashville, USA (2020, in press)*
18. Mehalik, M.M., Doppelt, Y., Schunn, C.D.: Middle-school science through design-based learning versus scripted inquiry: better overall science concept learning and equity gap reduction. *J. Eng. Educ.* **97**(1), 71–85 (2008)
19. National Academy of Engineering, National Academies of Sciences, Engineering, and Medicine: Science and engineering for grades 6–12: investigation and design at the center (2019). <https://doi.org/10.17226/25216>. <https://www.nap.edu/catalog/25216/science-and-engineering-for-grades-6-12-investigation-and-design>
20. National Research Council: *Engineering in K-12 Education: Understanding the Status and Improving the Prospects*. National Academies Press (2009)
21. National Research Council: *A Framework for K-12 Science Education: Practices, Crosscutting Concepts, and Core Ideas*. National Academies Press (2012)
22. NGSS Lead States: *Next generation science standards: for states, by states* (2013)
23. Pearl, J., Mackenzie, D.: *The Book of Why: The New Science of Cause and Effect*. Basic Books, New York (2018)
24. Penner, D.E.: Cognition, computers, and synthetic science: building knowledge and meaning through modeling. *Rev. Res. Educ.* **25**(1), 1–35 (2000). chap. 1
25. Pianta, R.C., Belsky, J., Houts, R., Morrison, F.: Opportunities to learn in America’s elementary classrooms. *Science* **315**(5820), 1795–1796 (2007)
26. Resnick, M., et al.: Scratch: programming for all. *Commun. ACM* **52**(11), 60–67 (2009)
27. Schreiber, J.B., Nora, A., Stage, F.K., Barlow, E.A., King, J.: Reporting structural equation modeling and confirmatory factor analysis results: a review. *J. Educ. Res.* **99**(6), 323–338 (2006)
28. Schwarz, C.V., et al.: Developing a learning progression for scientific modeling: making scientific modeling accessible and meaningful for learners. *J. Res. Sci. Teach. Official J. Natl. Assoc. Res. Sci. Teach.* **46**(6), 632–654 (2009)
29. VanLehn, K.: Model construction as a learning activity: a design space and review. *Interact. Learn. Environ.* **21**(4), 371–413 (2013)
30. Weintrop, D., et al.: Defining computational thinking for mathematics and science classrooms. *J. Sci. Educ. Technol.* **25**(1), 127–147 (2016)
31. Wilkerson-Jerde, M., Wagh, A., Wilensky, U.: Balancing curricular and pedagogical needs in computational construction kits: lessons from the DeltaTick project. *Sci. Educ.* **99**(3), 465–499 (2015)
32. Wing, J.M.: Computational thinking. *Commun. ACM* **49**(3), 33–35 (2006)
33. Wright, S.: On “Path analysis in genetic epidemiology: a critique”. *Am. J. Hum. Genet.* **35**(4), 757–768 (1983)
34. Zhang, N., Biswas, G., Chiu, J.L., McElhaney, K.W.: Analyzing students design solutions in an NGSS-aligned earth sciences curriculum. In: Isotani, S., Millán, E., Ogan, A., Hastings, P., McLaren, B., Luckin, R. (eds.) *AIED 2019. LNCS (LNAI)*, vol. 11625, pp. 532–543. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-23204-7_44