Computational Thinking in Support of Learning and Transfer

Nicole M. Hutchins, Vanderbilt University, nicole.m.hutchins@vanderbilt.edu,
Gautam Biswas, Vanderbilt University, gautam.biswas@vanderbilt.edu
Rachel C. Wolf, Stanford University, rcwolf@stanford.edu
Doris B. Chin, Stanford University, dbchin@stanford.edu
Shuchi Grover, Looking Glass Ventures, shuchig@cs.stanford.edu
Kristen P. Blair, Stanford University, kpilner@stanford.edu

Abstract: The role of transfer signifies a desire for learning activities to extend beyond the classroom setting to future problems and experiences. Efforts to evaluate the transfer of learned skills have aimed to better understand the value and broader impact of various learning activities. As key national and state standards emphasize the importance of integrating computational thinking (CT) into STEM classrooms, careful consideration should be made on how to teach and engage students in CT in a manner that supports the development of skills that can be applied in multiple contexts. Computational modeling integrated into science classrooms may support this goal. In this paper, we adopt a preparation for future learning approach to evaluate how proficiencies in CT are learned in a computational modeling curriculum and then transferred to problem solving in a new domain.

Introduction

Educators aim to provide students with knowledge and skills for future success. Coupled with the fast-paced nature of technological advancement, this provision entails the need to better prepare students to utilize technology and computational tools as vehicles for problem solving and professional advancement (Wing, 2011). The teaching of computational thinking (CT) skills in STEM (Science, Technology, Engineering, and Math) domains using a *learning by modeling* (e.g., Sengupta, et al., 2013) approach may support this goal. In this paper, we adopt a preparation for future learning (PFL; Bransford & Schwartz, 1999) perspective of transfer, proposing that *learning by modeling* will not only support learning of STEM and CT in a particular context, but will also support transfer of this learning into new STEM topics and domains.

In this study, we evaluate student PFL performance to measure students' conceptual understanding and their abilities to adapt and apply three key CT problem-solving skills: *decomposition*, *algorithmic thinking*, and *debugging* (Grover & Pea, 2018). Students completed *learning-by-modeling* tasks in two separate STEM domains. We first evaluate students' progress in the three CT skills using summative, formative, and modeling-task assessments to measure the development of student skill proficiency. These CT measures were then correlated with performance on a PFL assessment designed to capture student transfer of CT skills and practices in new domain and programming environment.

This paper primarily addresses the research question: *how does performance in CT skills and practices impact the ability to transfer problem-solving skills to new problems and contexts*? After a brief description of the Collaborative, Computational STEM (C2STEM) system, we present our methods, analyze the data collected from a classroom study to answer our research question, and discuss the implications of our findings and directions for future work.

Background

Traditional assessments rely heavily on the identification of deficiencies students may have compared to an expert, as opposed to determining what they are prepared to learn in the future (Schwartz, Bransford & Sears, 2005). Schwartz and colleagues (Bransford & Schwartz, 1999; Schwartz, Bransford & Sears, 2005) have proposed Preparation for Future Learning (PFL) assessments that utilize evidence from students' learning trajectories and current understanding to determine how they may construct new knowledge and solve new problems. In other words, PFL assessments adopt a double transfer paradigm (Schwartz & Martin, 2004). First, they identify what students "transfer in" to help them learn from a resource embedded in the assessment. Second, they determine what the students "transfer out" from the learning resource to correctly solve a new problem.

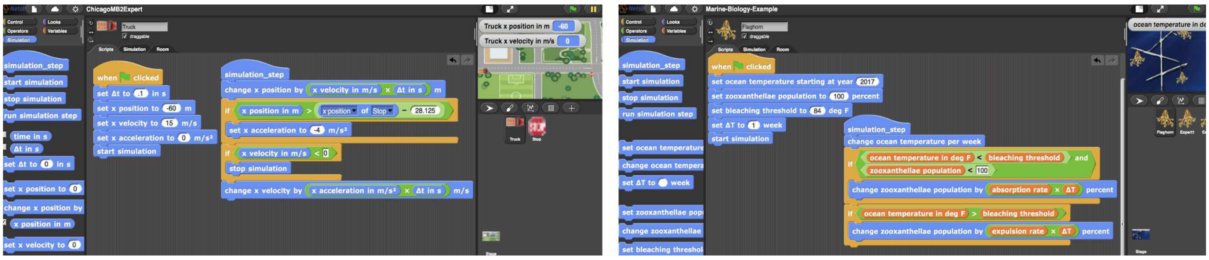
Researchers have found that ways in which a learning experience can support transfer include influencing the strategies and learning processes students may remember and deploy in the new context (Schwartz & Arena, 2013), and shaping students' understanding of specific concepts so that they may be better able to apply them later (Grover, Pea & Cooper, 2014). Few studies at the K-12 level have looked at the issue of PFL and transfer of CT skills to other STEM or computing contexts through designed PFL assessments (e.g. Chin, et al., 2010; Grover, Pea & Cooper, 2014; Schwartz & Martin, 2004). Given that there are several open questions in the field around the nature of transfer of CT skills between domains, and whether it even occurs (Denning, 2017), we believe that our work on examining which CT concepts and practices transfer across and support learning in STEM domains will help us gain a better understanding of the role of CT in STEM learning and instruction.

The C2STEM environment

We propose that a *computational modeling* approach provides a comprehensive framework for learning science and studying if the framework supports transfer of CT concepts and practices across science domains. Relevant CT skills and practices that we will study in this paper include: 1) *decomposition* to identify and initialize variables and their behaviors, 2) *algorithmic thinking* s using computational constructs; and 3) *debugging* and refining the generated model using information provided by the simulation environment.

Our computer-based learning environment, C2STEM, embodies a *learning-by-modeling* approach to support STEM and CT learning in the targeted domains of physics and marine biology. The system uses domain-specific modeling languages (DSMLs) to support students' model building, and its block-based, drag-drop interface allows for a low-entry threshold. We adopted evidence-centered design (ECD; Mislevy & Haertel, 2006) principles to align our complementary curriculum with NGSS (NGSS, 2013) and teachers' curricular needs. Students can build and execute their developing models of scientific phenomena, and the environment scaffolds their problem solving and *debugging* through simple animations, data tables, and graphing tools. The design and implementation of C2STEM is detailed in Hutchins, et al. (2019).

For each curriculum unit, students are scaffolded with a series of instructional and model-building tasks. Instructional tasks help students express relations between domain concepts as computational structures. Model-building tasks are more open-ended and require students to combine their STEM and CT knowledge to build specific scientific models (e.g., apply deceleration to slow a moving truck to make it stop at a STOP sign).



(a) (b)

Figure 1. Model-building tasks for physics (a) and marine biology (b) (visit C2STEM.org for tasks).

We have run multiple studies in middle school classrooms with C2STEM. In physics, students were tasked to model the deceleration of a truck initially moving at a constant velocity, and make it come to rest at a stop sign. In the physics context, (Figure 1a), students can initialize model variables (e.g., Δt, *x-position in m*,, *x-acceleration in* m/s^2) under the "*green_flag*" starting block (a construct used in Scratch and Snap!) and use constructs like "change *x-position* by [expression/value] m" in a loop (initiated by the "*simulation_step*" call block and flag) to model the relation between velocity and position of the truck in a *step-by-step* manner (each step represents Δt time units). We believe that this temporal *step-by-step* approach to modeling, in contrast to equation-based modeling, helps students learn to apply first principles to decompose the motion of the object into components (e.g., *update-velocity* and then *update-position* of an object at each time step) to gain a better general understanding of the dynamics of motion. In marine biology (Figure 1b), students apply a similar *step-by-step* approach to building models of coral growth and bleaching as a function of ocean temperatures. Students worked with real ocean temperature data from 2016-2017 that captured a mass bleaching event along the Great Barrier Reef. The primary relations that the students modeled were the expulsion and absorption rates of zooxanthellae algae (this gives algae their color) as a function of ocean temperature.

A previous study in a semester-long high-school physics classroom showed that students who participated in our C2STEM curriculum were more likely to apply general problem-solving strategies in a new physics context on a PFL assessment than students in a traditional physics classroom (Hutchins, et al., 2019). However, more research is needed to better understand the thresholds that constitute the "*sufficient degree of original learning*" (Bransford & Schwartz, 1999) to support transfer of problem-solving skills. More specifically, we need to better understand how students acquire the relevant CT skills when building computational models, and how these skills support better learning and problem solving in future contexts.

Methods

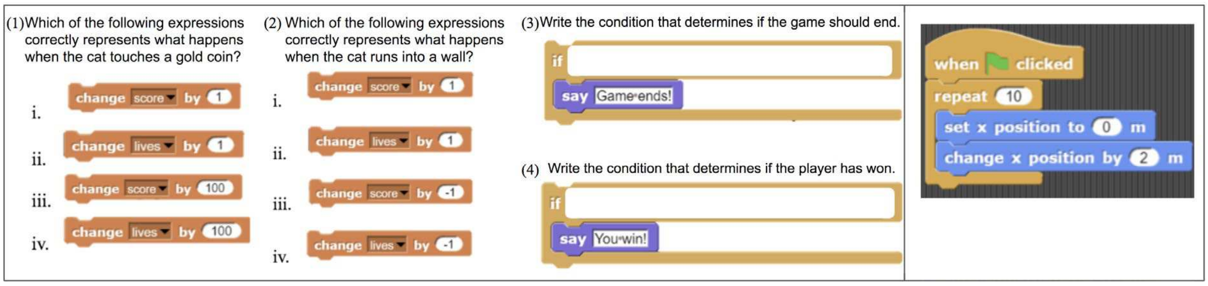
In the current study, 43 middle-school students completed a series of instructional and model-building tasks focused on one-dimensional motion with acceleration using C2STEM for 4 class days (excluding the time for summative and PFL assessments). This included a 45-minute training task centered on introducing students to the environment and its key components. There was a three-month gap before students again used C2STEM for a week to learn marine biology. To the best of our (and the teacher's) knowledge, students were not engaged in coding during that break in any class at the school.

Data measures

Student learning was measured using: (1) summative pre-post-tests; (2) two formative assessment tasks embedded in the curriculum; (3) model-building tasks; and (4) a post-curriculum PFL assessment.

Summative and formative assessments

We used a pre-post design to measure students' conceptual knowledge in the STEM domains and CT. The physics instrument included 8 Physics questions (maximum: 23 points) and 4 CT questions (maximum: 22 points) that were drawn from prior research (Grover, 2020; Hutchins, et al., 2019). The marine biology instrument included 7 items (maximum: 13 points) and the same CT questions except that the physics DSML blocks were substituted by marine biology-DSML blocks. Figure 2 provides examples of CT questions to be discussed.



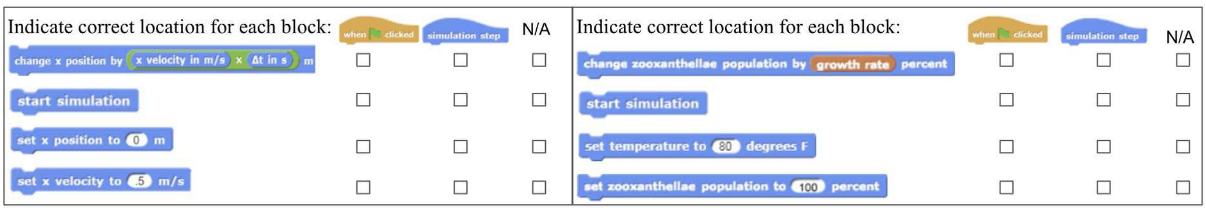
(a) (b)

Figure 2. Summative assessment questions in algorithmic thinking (a) and debugging (b).

We evaluated students' learning gains in *decomposition*, *algorithmic thinking*, and *debugging* using three items from the CT summative assessment. The first item (from Grover, 2020) tests *decomposition*. Students were given a description and rules for a maze game. During the game, a cat collects coins to increase the player's score, but the score decreases if the cat touches a wall. Students were asked to list two variables needed to program the game, describe the behavior for each variable, and set the variable's initial value. Following this, a series of questions, tested their *algorithmic thinking* skills (Figure 2a), i.e., students' understanding of the steps that govern the running of the program.

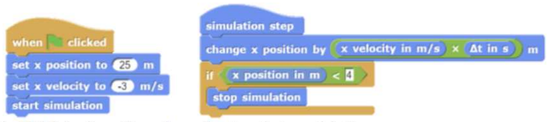
The final question targeted *debugging* (Figure 2b). Students were given a block of code that simulated the motion of a ball starting at position = 0 and traveling to the right for 20 meters. Students had to describe the ball's motion based on the code, and then describe how they would fix the code to correctly simulate the motion of the ball.

Decomposition and *algorithmic thinking* were also evaluated in embedded formative assessments. The physics and marine biology questions were matched, assessing similar constructs at similar time points in the two curricula. Figures 3 and 4 shows check-in 1 and 2, respectively, for physics and marine biology. It is important to note that check-in 2 (Figure 4) required more advanced CT skills. For instance, the *algorithmic thinking* questions require students to create a variable construct to track model changes over time (e.g., the total distance traveled).



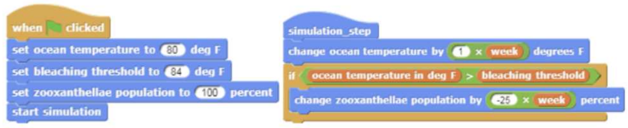
(a) (b)

Figure 3. Check-in 1 in physics (a) and marine biology (b).



A. Which direction does the truck travel in?
B. What is the truck's speed while in motion?
C. What is the total distance that the truck will have traveled when the simulation ends?
D. For how much time will the truck have traveled when the simulation ends?

(a)



A. What is the bleaching threshold equal to?
B. When will the zooxanthellae population change?
C. How much does the ocean temperature change per year?
D. How long will it take for the zooxanthellae population to equal 0%?

(b)

Figure 4. Check-in 2 in physics (a) and marine biology (b).

Assessing model-building tasks

Model-building and problem-solving tasks in physics were presented as transportation problems with the purpose of delivering medicine (via land, water, and air) from a port city to a remote village. In marine biology, students performed similar instructional and model-building tasks to study coral growth and decay. Their goal was to construct models to understand the impact of climate change on coral reefs. (Details of these tasks can be found at http://c2stem.org). All modelling tasks were scored using a predefined rubric (adapted from Hutchins, et al., 2019) that are consistent across units. Example rubric items are presented in Table 1.

Preparation for Future Learning (PFL) assessment

PFL assessments for C2STEM aimed to capture how a CT-based approach would help foster problem-solving strategies and conceptual understanding in other STEM contexts. Students were given a modified NetLogo model simulating the flocking of birds and a set of rules that governed the behaviors of the birds (see http://ccl.northwestern.edu/netlogo/). Students had to: (1) explore the model and describe three specific things they understood from the model (PFL Task 1); (2) describe and reason about how they would change the given model to simulate the behavior of hypothetical dragons that typically exhibit separation as opposed to flocking behaviors (PFL Task 2). We analyzed students' ability to decompose the original model to identify relevant variables and their effects on system behavior, as well as their accuracy in updating the model given the need to hypothetical behavior of dragons. Scores on the two tasks were summed to give an overall PFL performance score that ranged from 0 to 8.

Table 1 presents our scheme for measuring CT performance in *decomposition*, *algorithmic thinking*, and *debugging* during the course of the two *learning-by-modeling* units and expected transferred behaviors based on a demonstration of those CT skills.

Table 1: Scheme for studying PFL and transfer of CT skills through *learning by modeling* in C2STEM

	Determining CT Performance During Curriculum		**Determining CT Transfer**
CT Construct	Formative & Summative Assessments	Model-Building Rubric	PFL Assessment
Decomposition	List and describe variable(s) needed for a program Correctly implement block under either green flag or simulation step (e.g., Figure 3)	Separate initialization ("set" block under green flag) from updating ("change" block under simulation step) variables	List parameters needed for model in a new domain
Algorithmic Thinking	Correctly change variable based on problem description (e.g., Figure 2a, Questions 1 & 2) Implement conditional logic based on desired behavior (e.g., Figure 2, Questions 3 & 4)	"Set" or "change" variables: (1) in the correct order, (2) in the correct sequence, (3) under correct conditions	Describe how changing parameter values impacts model behavior

Debugging	Locate and describe error in code based on problem description (e.g., Figure 2b) Describe how to correct given code to appropriately solve problem	Play model to receive visual feedback about its behavior Locate and correct errors based on evidence (visual or data) compared to expected model behavior	Play model to apply *decomposition* skill Identify elements of model needed to change model behavior based on new problem description

Results

We discuss our analyses to answer the core research question, *does the level of proficiency in CT skills impact the ability to transfer problem-solving skills to new domains?* It is important to note that severe weather affected the last day of our study, and the classroom teacher made the marine biology posttest a take-home assignment. We have not used this posttest data in our analysis. In addition, students only had 20 minutes to work on the PFL assessment due to scheduling constraints. Summative assessment results from the physics unit showed significant learning gains in STEM and CT. This result parallels other research demonstrating that *learning by modeling* supports STEM learning (Sengupta, et al., 2013; Klopfer, Yoon & Um, 2005). For this paper we will focus on CT performance.

CT learning from physics to marine biology

Table 2 reports student performances on the formative assessments. Differences between units are examined using t-tests. Although we did not observe significant improvements from physics to marine biology on the initial check-in, student performance on the second check-in (targeting *decomposition* and *algorithmic thinking*) significantly improved during the marine biology intervention. For the instructional and model-building tasks, CT scores showed significant improvements from physics to marine biology.

Table 2: Improvements in learning from physics to marine biology (∗: p-value < 0.05, ∗∗: p-value < 0.01)

Item	**Physics Unit Mean (SD)**	**Marine Biology Unit Mean (SD)**	***p* value**	**effect size**	**n**
Check-In 1	0.79 (0.35)	0.82 (0.21)	0.65	0.05	33
Check-In 2	0.38 (0.19)	0.76 (0.18)	<0.0001**	0.72	32
Instructional Tasks CT Score	0.80 (0.15)	0.91 (0.10)	0.001**	0.39	30
Model-Building Tasks CT Score	0.75 (0.21)	0.90 (0.18)	0.04*	0.35	14

CT constructs and future learning

Given our hypothesis that the development of CT-inspired problem-solving skills will support the transfer of problem-solving skills between science domains, we predict that students who demonstrated higher summative CT assessment scores would also perform better on PFL assessments. We ran an ordinary linear regression (OLR) with average summative performance scores (standardized) for each target CT construct (*decomposition*, *algorithmic thinking*, and *debugging*) as independent variables and the PFL score as the outcome measure. Best-fit model parameters show that each of the CT construct scores predicted the PFL score. *Decomposition* (R^2 = 0.758, F(1, 36) = 112.899, p < .000) proved to be the strongest predictor of PFL performance (β = 0.871, *p value* = 0.000). *Debugging* (R^2 = 0.323, F(1, 32) = 15.269, p < .000) and *algorithmic thinking* (R^2 = 0.249, F(1, 32) = 15.269, p < .003) also demonstrated significance (β = 0.568, *p value* = 0.000 and β = 0.499, *p value* = 0.003, respectively). The results support our original hypothesis that the learning of CT constructs supports future learning in a new STEM context. To gain a deeper understanding of the relations between the PFL measures and the CT skills we performed pairwise correlation analysis.

Correlation with Preparation for Future Learning assessment

Table 3 reports the correlation coefficients (Spearman's rho) of the PFL task with standardized performance measures. PFL results strongly correlated with the physics pre-posttest. However, the high pretest correlations indicated that prior CT knowledge may have also had an impact. Deeper analysis of individual questions also indicated that the learning of the target CT constructs may impact the ability to transfer across science domains. As seen in Table 3, it is interesting to note that correlations with *decomposition* and *debugging* peaked when measured after physics-modeling instruction. *Decomposition* and *algorithmic thinking* performance during marine biology were strongly correlated with PFL score, even after the significant break in between units. It is also worth noting that correlations with students' scores on *algorithmic thinking* were strongest when assessed during later check-in (formative) assessments. This result is worth exploring in future analyses.

We similarly analyzed results from the formative assessment in terms of the target CT constructs. We focus on check-in 2, which involved more complexity and advanced CT (e.g., the ability to create a variable to map its behavior over time to answer a question). PFL performance showed significant correlations with the *decomposition* component of check-in 2 for physics and the *algorithmic thinking* component of the final check-in—marine biology check-in 2. Note there was a ceiling effect for the *decomposition* component of marine biology check-in 2, and we, therefore, would not expect a correlation with PFL results. Check-in 2 also showed overall improvements (see Table 2) from the physics to marine biology units. The physics check-in 2 *algorithmic thinking* score showed the low overall performance score with little variability, which may explain the PFL correlation. This result suggests improved performance in CT practices, particularly *algorithmic thinking*, over time. Combined with the pattern of correlations between PFL performance and *algorithmic thinking*, we believe that this construct may have needed a longer time for students to develop the necessary skills to potentially transfer. We believe that improved understanding of the *decomposition* process in terms of *step-by-step* modeling across domains (units 1 and 2) also supports transfer of this practice to new domains. These results coincide with our findings in Hutchins, et al. (2019) in which the experimental group was better able to break down a continuous process (equation of motion) into a *step-by-step* discrete formulation of the equations.

Table 3: Summative and check-in assessment scores correlated with PFL (∗: p-value < 0.05, ∗∗: p-value < 0.01)

		Correlation Between PFL Performance and CT Scores		
Item		Decomposition	Algorithmic Thinking	Debugging
CT Pre-Test: Physics Unit	Correlation Coefficient	.577**	.334	.607**
	Sig. (2-tailed)	.001	.077	.000
	N	29	29	29
CT Post-Test: Physics Unit	Correlation Coefficient	.731**	.139	.740**
	Sig. (2-tailed)	.000	.471	.000
	N	29	29	29
CT Pre-Test: Marine Biology Unit	Correlation Coefficient	.399*	.471**	.333
	Sig. (2-tailed)	.026	.008	.072
	N	31	31	30
Check-in 1: Physics Unit	Correlation Coefficient	.035	.223	
	Sig. (2-tailed)	.841	.191	
	N	36	36	
Check-in 2: Physics Unit	Correlation Coefficient	.351*	-.118	
	Sig. (2-tailed)	.036	.495	
	N	36	36	
Check-in 1: Marine Biology Unit	Correlation Coefficient	.174	.100	
	Sig. (2-tailed)	.366	.605	
	N	29	29	
Check-in 2: Marine Biology Unit	Correlation Coefficient	.063	.436*	
	Sig. (2-tailed)	.742	.016	
	N	30	30	

Finally, an analysis of the rubric scores for the *decomposition* and *algorithmic thinking* components of students' final model-building code did not correlate PFL performance. As the constructs we are targeting involve processing of information, we hypothesize this result may be due to the use of final code instead of the processes students implemented to get to the solutions. An example of process-oriented behavior that will be the focus of future analysis is exemplified in our *debugging* case study.

Case study: *Debugging* in action

The following case description contrasts the approach to *debugging* by a student who had a high PFL score versus one with a low PFL score. StudentA demonstrated above-average normalized learning gains in CT. The student scored a 4.5 on the PFL (also above average). StudentB demonstrated below-average normalized learning gains in CT. StudentB scored a 1.5 on the PFL assessment. This case study will examine *debugging* episodes during a model-building task in which students needed to simulate a truck slowing down to a stop sign given a maximum deceleration and the *x*-position of the stop sign.

StudentA's need for *debugging* was caused when they modified their code to incorporate acceleration. The student removed the block to set an initial value of x velocity and added the block to set the x acceleration, setting it to the speed limit value of 15 (as seen in Figure 5, StudentA Step 1). This caused the x-velocity to be initialized to 0 m/s (the automatic system setting). To debug the issue, StudentA used visual feedback from the simulation in a systematic way (e.g., the running of the simulation only 1 or 2 times may indicate checking of specific computational blocks), a rubric item from Table 1. At each Step in Figure 5, StudentA ran (or played) the simulation to get visual feedback of the changes made to the code. Following the code shown in StudentA's Step 4 (Figure 5), the student went on to add the change velocity block and update appropriately. These *debugging* skills extended in error identification and correction during the Marine Biology unit indicating that the student used the environment in a productive way to pinpoint an error in the model and correct it.

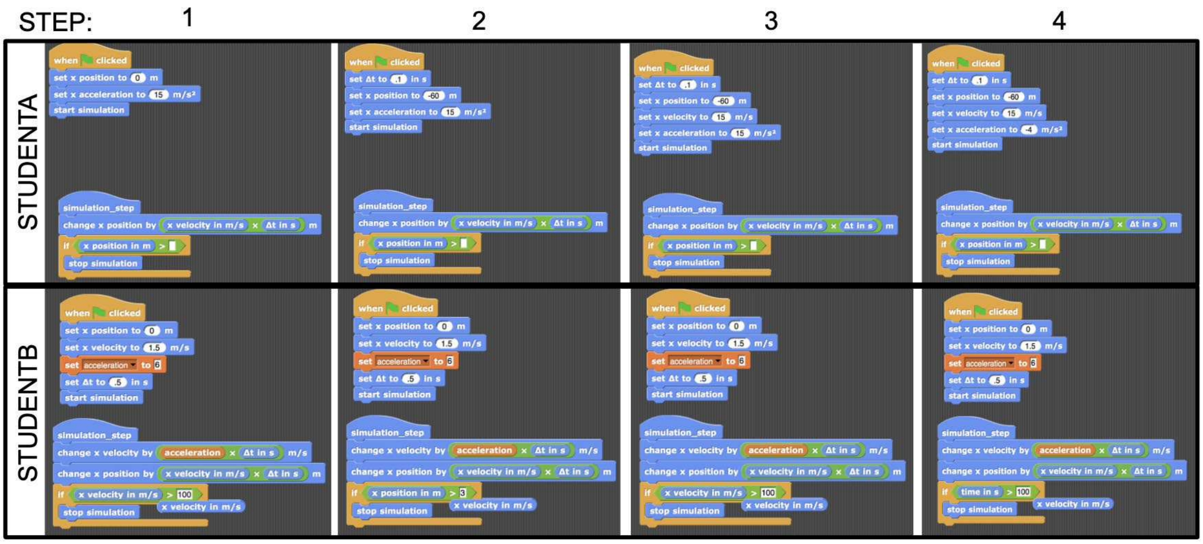


Figure 5. Two students' *debugging* processes in C2STEM.

StudentB's error is located in their stopping condition illustrated in Figure 5, StudentB, Step 1 (it did not incorporate the x-position of the stop sign, nor any coordinate on the stage). In trying to correct these errors, the student adopted a trial-and-error approach to debug the model. After Step 1, the student played the simulation three times, opened the data table, and played the simulation again. Due to the condition that the simulation would stop when the x-velocity was greater than 100, the student was unable to see if the condition worked as the truck was not viewable on the stage. However, StudentB closed the table and edited the code to the image shown in StudentB, Step 2 (Figure 5). This model also did not support appropriate visual feedback as the simulation step would only run once based on its conditional logic. The student ran the simulation once, and then edited the code to Step 3. The student did this once more, could not correct the errors, and gave up.

This case study shows that StudentA was able to utilize various simulation data (visualization on the stage, the DSML blocks, and the data tools) to identify and correct errors in their code. As this student not only demonstrated strong CT gains throughout, but also above-average scores on the PFL, we conjecture that the student was able to transfer *debugging* skills (e.g., the evaluation of model feedback to understand model decomposition and behavior) to support their PFL performance. However, further analysis of students' real-time *debugging* activities may help us to better understand how CT practices transfer across problem domains.

Discussion and implications

The call to engage students in authentic science experiences provides an impetus for creative curricular design that combines STEM and CT learning that, in turn, supports preparation for future learning. Our systematic approach to understanding how learning of CT may impact student abilities to transfer problem-solving skills to new domains has demonstrated the importance of multiple assessments through different stages of an intervention to understand how CT constructs and practices are learned over time. These results also may indicate the importance of these constructs in supporting learning and problem solving in multiple domains. Our results provide preliminary evidence on how development of CT skills (specifically, *decomposition*, *algorithmic* *thinking*, *debugging*) supports transfer of problem-solving skills to new domains. The paper also sheds light on *decomposition* and *debugging* as key transferable skills. Although *algorithmic thinking* was not as strong a predictor of PFL performance, final formative assessment performance did correlate with PFL performance. We conjecture that students did not achieve sufficient original learning (short study time) for all of the targeted CT practices.

We recognize the limitations in our work, particularly those related to study logistics. For instance, timing played a role in performance. There was a significant time difference between the two *learning-by-modeling* units and we could not collect data on student programming experience outside of the school environment. In addition, our PFL results may have been impacted by the short timeframe provided for students to complete this type of assessment. These issues will be corrected in the next version of this study with changes to the duration of the study and the collection of process data. Finally, in order to more thoroughly examine the learning thresholds of *decomposition*, *algorithmic thinking*, and *debugging,* we have enhanced our summative and formative assessments with more robust items (including the addition of *debugging* problems).

Despite these limitations, our PFL approach to the evaluation of student learning and transfer during *learning by modeling* provides a novel way to measure how transferable problem-solving skills are acquired and applied in new contexts. This paper provides a framework for the study of how CT skills transfer and demonstrates how the acquisition of CT skills may predict future learning in integrated STEM and computing contexts.

References

Bransford, J. D., & Schwartz, D. L. (1999). Chapter 3: Rethinking transfer: A simple proposal with multiple implications. *Review of research in education*, *24*(1), 61-100.

Chin, D. B., Dohmen, I. M., Cheng, B. H., Oppezzo, M. A., Chase, C. C., & Schwartz, D. L. (2010). Preparing students for future learning with Teachable Agents. *Educational Technology Research and Development, 58*, 649–669.

Denning, P. (2017). Remaining Trouble Spots with Computational Thinking. *Communications of the ACM, 60*(6), 33-39.

Grover, S., Pea, R. (2018). Computational Thinking: A competency whose time has come. In Sentance, S., Carsten, S., & Barendsen, E. (Eds). *Computer Science Education: Perspectives on teaching and learning*, pp. 19-38. Bloomsbury.

Grover, S. (2020). Designing an Assessment for Introductory Programming Concepts in Middle School Computer Science. In *Proceedings of the 51st ACM Technical Symposium on Computing Science Education (SIGCSE'20),* Portland, OR.

Grover, S., Pea, R. and Cooper, S. (2014). Expansive Framing and Preparation for Future Learning in Middle-School Computer Science. In *Proceedings of the 11th International Conference of the Learning Sciences*.

Hutchins, N., Biswas, G., Maróti, M., Lédeczi, A., Grover, S., Wolf, R., Blair, K., Chin, D., Conlin, L., Basu, S., & McElhaney, K. (2019). C2STEM: a System for Synergistic Learning of Physics and Computational Thinking. *Journal of Science Education and Technology*, 1-18.

Klopfer, E., Yoon, S., & Um, T. (2005). Teaching complex dynamic systems to young students with StarLogo. *Journal of Computers in Mathematics and Science Teaching, 24*(2), 157–178.

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

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

Schwartz, D. L., & Arena, D. (2013). *Measuring what matters most: Choice- based assessments for the digital age*. Cambridge, MA: MIT Press.

Schwartz, D. L., Bransford, J. D., & Sears, D. (2005). Efficiency and innovation in transfer. In J. Mestre (Ed.), *Transfer of learning: Research and perspectives* (1–52). Greenwich, CT: Information Age Publishing.

Schwartz, D. L., & Martin, T. (2004). Inventing to prepare for future learning: The hidden efficiency of encouraging original student production in statistics instruction. *Cognition and Instruction, 22*(2), 129–184.

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.

Wing, J. (2011). Research notebook: Computational thinking—What and why. *The Link Magazine*, 20-23.

Acknowledgments

This research is supported by NSF grant #1640199.