

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/327045211>

A principled approach to designing assessments that integrate science and computational thinking

Conference Paper · June 2018

CITATIONS

0

READS

148

5 authors, including:



Satabdi Basu

SRI International

26 PUBLICATIONS 255 CITATIONS

[SEE PROFILE](#)



Shuchi Grover

45 PUBLICATIONS 742 CITATIONS

[SEE PROFILE](#)



Gautam Biswas

Vanderbilt University

552 PUBLICATIONS 7,490 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Intelligent Distributed Diagnosis And Prognostics For Hybrid Systems, IDEAPHYS [View project](#)



Adaptive Reflexive Resource Management System [View project](#)

A principled approach to designing assessments that integrate science and computational thinking

Satabdi Basu, SRI International, satabdi.basu@sri.com
Kevin W. McElhaney, SRI International, kevin.mcelhaney@sri.com
Shuchi Grover, shuchig@cs.stanford.edu
Christopher J. Harris, SRI International, christopher.harris@sri.com
Gautam Biswas, Vanderbilt University, gautam.biswas@vanderbilt.edu

Abstract: There is increasing interest in broadening participation in computational thinking (CT) by integrating CT into precollege STEM curricula and instruction. Science, in particular, is emerging as an important discipline to support integrated learning. This highlights the need for carefully designed assessments targeting the integration of science and CT to help teachers and researchers gauge students' proficiency with integrating the disciplines. We describe a principled design process to develop assessment tasks and rubrics that integrate concepts and practices across science, CT, and computational modeling. We conducted a pilot study with 10 high school students who responded to integrative assessment tasks as part of a physics-based computational modeling unit. Our findings indicate that the tasks and rubrics successfully elicit both Physics and CT constructs while distinguishing important aspects of proficiency related to the two disciplines. This work illustrates the promise of using such assessments formatively in integrated STEM and computing learning contexts.

Introduction

Driven by the needs of a 21st century workforce, education and industry stakeholders recognize that computing knowledge and skills provide the foundation for competency in a multitude of fields (Wing, 2006). One approach for making computational thinking (CT) accessible to K-12 (e.g. formal precollege) students is to integrate it with existing components of the K-12 Science, Technology, Engineering, and Mathematics (STEM) curricula. STEM topics lend themselves particularly well to integration with CT, because many of the epistemic and representational practices central to expertise in STEM disciplines (e.g., characterizing problems and designing solutions, developing and using models, analyzing and interpreting data) are also primary components of CT proficiency (Basu et al., 2016). The integration of STEM and CT in K-12 settings is further motivated by current STEM workforce practices that increasingly rely on computational modeling and simulation tools for understanding, analyzing, and solving problems (Landau, 2006; Freeman et. al., 2014). The US Framework for K-12 Science Education (NRC, 2012) and Next Generation Science Standards (NGSS Lead States, 2013) instantiate this view by including 'Using Mathematics and CT' as a key science and engineering practice.

With computer science frameworks (e.g., the US K-12 CS Framework, 2016) gaining traction in K-12 instructional settings, science will likely emerge as an important context for teaching CT in school. Leveraging the synergy between CT and science in K-12 classrooms will require, among other things, the systematic development of assessments that measure learning at the intersection of science and CT. These assessments will need not only to integrate the science and CT disciplines, but also integrate disciplinary concepts and practices, following the vision put forth by contemporary STEM education frameworks that integrate content and practice.

In this paper, we describe a general principled approach for designing rich assessment tasks and associated rubrics that integrate science disciplinary knowledge, CT concepts, and computational modeling practices using Evidence Centered Design (ECD) principles (Mislevy & Haertel, 2006). We discuss an application of this approach where we designed and administered multiple assessment tasks embedded within a web-based, computational modeling environment that supports the integrated learning of physics and CT for high school students. Using one such task and an associated rubric as an example, we use video recordings of students responding to the task to analyze students' responses to the task, illustrating (1) how the task elicits different aspects of students' science (physics) and CT proficiencies in this integrated domain and (2) how the rubric distinguishes these aspects of proficiency for the purposes of formative assessment.

Theoretical perspectives

Synergistic learning of science and CT

Developing a computational model of a physical phenomenon involves integrating key aspects of CT and scientific practice: identifying appropriate abstractions (e.g., underlying rules governing the behavior of relevant

entities), making iterative comparisons of the generated representations with the target phenomenon, and debugging the abstractions to generate progressively sophisticated explanations of the phenomenon. Numerous research studies have shown that integrating CT and scientific modeling can be beneficial (e.g., Hambrusch et al., 2009; Blikstein & Wilensky, 2009; Basu, Biswas & Kinnebrew, 2017). Sengupta et. al (2013) describe how integrating CT and scientific modeling can be beneficial: (1) Lower the learning threshold for science concepts by reorganizing them around intuitive computational mechanisms: computational representations introduce discrete and qualitative forms of fundamental laws, which are simpler to understand than equation-based continuous forms (Redish & Wilson, 1993); (2) Programming and computational modeling as representations of core scientific practices: Soloway (1993) argued that learning to program amounts to learning how to construct mechanisms and explanations; and (3) Contextualized representations make it easier to learn programming (Papert, 1991). These benefits reflect the framing of proficiency in both science and CT (by the NGSS and K-12 CS Framework, respectively) as the integration of knowledge and practice.

Evidence-Centered Design

We use ECD, a principled assessment design framework (Mislevy & Haertel, 2006), to create assessments that are inclusive of science and CT concepts and practices. ECD promotes coherence in the design of assessment tasks and rubrics and the interpretation of students' performances by explicitly linking claims about student learning, evidence from student work products, and design features of tasks that elicit the desired evidence. ECD begins with a domain analysis, which entails gathering and organizing information on the domain to be assessed. This is followed by domain modeling, which entails the articulation of specific learning targets and task design specifications, which in turn inform the development of tasks and rubrics. ECD has been used to develop CS assessments for the Exploring Computer Science curriculum (Goode et al., 2012), as well as to develop science assessments that integrate content knowledge with science practices along the performance dimensions of the NGSS for summative and formative purposes (Harris et al., 2016).

Methods

Designing synergistic assessment tasks for measuring science and CT proficiencies

Figure 1 illustrates an ECD process for creating assessments that are inclusive of science and CT, while targeting concepts and practices that cut across science and CT. Our process begins with identifying the integrated science and CT domain, for example 'High school kinematics and CT', or 'Middle school carbon cycle and CT'. Then, in the domain analysis phase, we unpack the three domains of science disciplinary concepts, CT concepts, and computational modeling practices. We elaborate on and document the target constructs that we want to assess in each domain, determine assessment boundaries and expected background knowledge for the domains, and articulate the knowledge, skills, and abilities (KSAs) relevant to each domain. Next, we create integrated domain maps to represent the relationships and synergies between the three domains. These maps are important because they enable us to be principled in our choice of which science and CT concepts and modeling practices to integrate in an assessment task. The integrated domain maps offer a range of ways to coherently express integrated learning goals for science and CT during the domain modeling phase.

The integrated learning goals constitute the claims we make about what students should know and be able to do. Additionally, for each learning goal, we articulate a design specification that guides the design of tasks and rubrics aligned to it (Mislevy & Haertel, 2006). Each design specification focuses on the following aspects that provide the basis for tightly integrating task and rubric design: (1) focal KSAs, (2) features of student responses that constitute evidence of proficiency with each focal KSA, (3) characteristic features of assessment tasks that can effectively elicit this evidence of proficiency, and (4) variable task features that can shift the difficulty or focus of a task. In the task and rubric development phase, these design specifications and technology affordances of the task delivery system inform the development of tasks and rubrics in a way that aligns the assessment targets, desired evidence of student proficiency, task design features, and scoring criteria. Though the design process may appear to be linear and unidirectional, it is iterative in nature, allowing developed tasks to help refine the learning goals or design specifications, for example.

Applying the approach described in Figure 1, we have developed multiple assessment tasks to measure high school students' integrated proficiencies in Physics (high school kinematics) and CT. An initial domain analysis helped us identify a set of target constructs for each of the domains of physics disciplinary concepts, CT concepts and computational modeling practices. Based on these constructs, we articulated a set of learning goals, each integrating physics concepts, CT concepts, and an aspect of computational modeling practice.

Figure 2 illustrates selected constructs that we identified in each domain, as well as a few sample learning goals that we articulated by integrating the constructs. For example, the first learning goal articulated in Figure 2

(in boldface) integrates target constructs from all three domains (in boldface). Before articulating learning goals, we created integrated domain maps where we identified key relationships between the physics and CT domains to ensure that the integration of the physics and CT concepts for each learning goal leveraged the synergy between the domains (instead of combining physics and CT concepts arbitrarily). For example, calculating the velocity of an object based on its initial velocity, acceleration and time closely relates to the CT concepts of initializing and updating variables (velocity, acceleration and time are all examples of variables), and operators and expressions. Additionally, combining the related physics and CT concepts with different aspects of computational modeling practices like ‘Develop, Use, Test, Debug’ helped create learning goals that guided task design specifications at different levels of complexity.

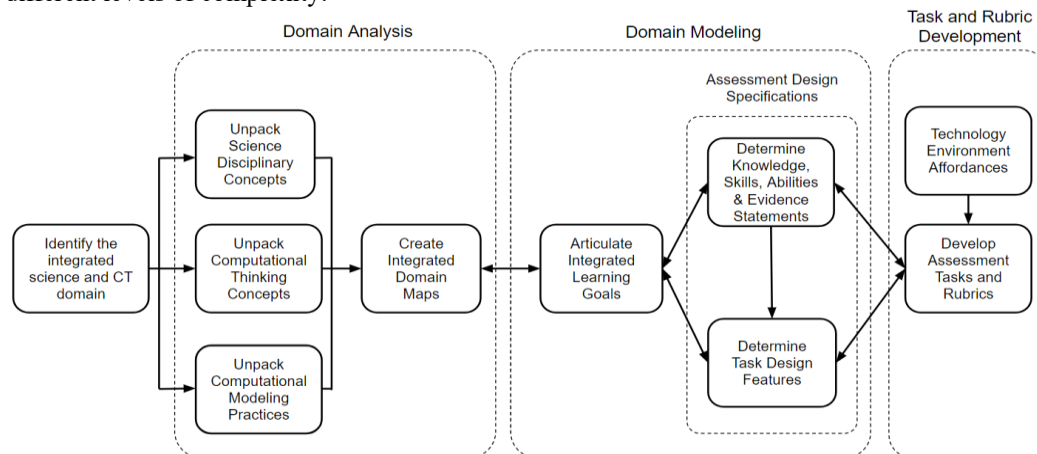


Figure 1. Design process schematic for assessment tasks that integrate science and CT learning.

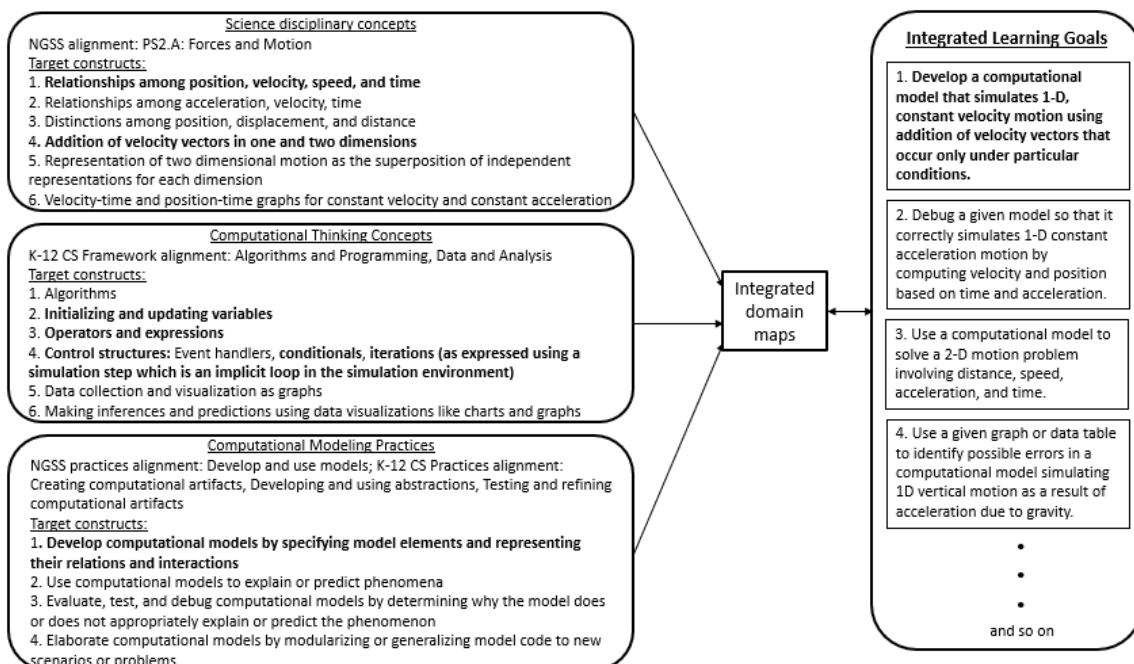


Figure 2. Unpacking the physics and CT domains, identifying their relationships through integrated domain maps, and the articulation of integrated learning goals. (Bold text illustrates how a learning goal integrates concepts and practices across disciplines).

Based on the learning goals, we developed 18 tasks of varying complexity comprising various formats such as multiple choice, explanation, and programming. In some tasks, we provided most of the code and asked students to fill in a small part that targeted a specific concept, while in other tasks, we provided required blocks and asked students to focus only on arranging the blocks in a correct computational sequence. We created different versions of debugging tasks such as asking students to correct a given buggy program; showing students a

snapshot of a program and asking them to indicate which block(s) to modify and how; and asking students to use resultant data and graphs to identify errors in a hypothetical program not shown to them.

Empirical study using assessments to elicit integrated science and CT proficiencies

We embedded the assessment tasks in the C2STEM learning environment – a browser-based system that engages students in computational modeling and simulation of Physics phenomena. The computational modeling representation uses custom domain-specific blocks developed on top of NetsBlox (Broll et. al., 2016), a block-based extension of Snap! (<http://snap.berkeley.edu/>) to help learners focus on physics concepts.

We conducted an empirical pilot study to examine how well our assessment tasks elicited students' proficiencies in integrating physics and CT, and how rubrics could be designed to distinguish between components of proficiency across students. The study was conducted within a high school summer program for Science and Math. The students worked on three C2STEM modules as part of a 10-hour kinematics curriculum, with each module comprising an alternating sequence of scaffolded modeling activities and embedded assessments. All of the participating high school students had prior experience working with NetsBlox as part of prior summer school activities, and some reported familiarity with languages like Scratch and Python, but none had taken a high school physics class.

In this paper, we limit our analyses to one assessment task, the Airport task (Figure 3), which addresses the learning goal *'Develop a computational model that simulates 1-D, constant velocity motion using addition of velocity vectors that occur only under particular conditions.'* We examine 10 students' responses (4 female, 6 male) to this task to determine how well it elicits evidence for target physics and CT constructs in the context of computational modeling, and also differentiates among levels of proficiency within the domains.

The screenshot displays the 'Airport Task' assessment interface. At the top, a navigation bar shows 'Home > 2D: Water > Check-in #1' and 'ASSESSMENT TASK_2.1'. Below this, 'Question 1.1' is presented with a text description of the physics problem involving Josh and Kate walking through an airport with a moving walkway. A diagram illustrates the setup with points A, B, and C, and a moving walkway between B and C. Below the diagram, three bullet points provide instructions for the student: to open the code for 'Airport Task', to complete a procedure for setting Josh's resultant velocity, and to stop the simulation when Josh reaches Point C. The bottom portion of the image shows the NetsBlox code editor with various physics-related blocks, including 'when clicked', 'set position to x: y: z:', 'set Josh's speed to', 'simulation step', and 'update position'.

Figure 3. The “Airport Task: An example programming assessment task.

Data sources and plans for analyses

We recorded all student responses to assessment tasks using the Camtasia™ screen-capture software. We examined the screen recordings to characterize students' model-building approaches and challenges faced. We

noted whether students solved the tasks correctly on their first attempts or whether they required multiple iterations of testing and debugging. For students submitting an incorrect solution, we recorded the different types of errors and verified that the students made an honest attempt to solve the tasks. Based on the analysis, we developed a rubric (Table 1) that scores students' final programming solutions (not their model-building approaches) along two aspects of integrated physics-CT proficiency: (1) the ability to express physics relations in a computational model and (2) the ability to use programming concepts to model a physics phenomenon. Scoring the task based on these distinct aspects of proficiency has the potential to provide useful information to researchers and teachers on the specific nature of students' proficiencies.

Table 1: Rubrics for characterizing student performance on an integrative assessment task

Criterion #	Rubric for scoring the Airport task	Points
Expressing physics relations in a computational model (physics component): 2 point rubric		
1	Program expresses correct relations among velocity, position and time, and correct units for each	1 point
2	Program reflects that walking on the moving walkway causes resultant speed to be additive in the x direction (walking speed +walkway speed) and constant (no acceleration)	1 point
Using programming concepts to model physics phenomena (CT component): 4 point rubric		
3	Program makes the distinction between actions that need to happen once during initialization and actions that need to be repeated in the simulation step	1 point
4	Program correctly determines which action always happens and which happens under certain conditions	1 point
5	Program updates the variable corresponding to Josh's velocity on the walkway <ul style="list-style-type: none"> a. under the correct conditions (Use conditionals with appropriate expressions to update Josh's velocity under correct conditions between Point B and Point C only), and b. in the correct fashion (the x velocity is set to a new constant value instead of changing at every simulation step) 	1 point
6	All code in the program is reachable and can be executed	1 point

Findings

Scoring students' final programming artifacts for the Airport task using the rubric described in Table 1 revealed that half the students (s1 through s5) solved the task correctly, earning the maximum scores (2 and 4, respectively) on the physics and CT rubric components. Among the students who were unable to solve the task correctly, some students (s6 – s8) demonstrated high proficiency with the physics component (scoring 2 points), but only partial proficiency on the CT component (scoring less than 4 points). Two others (s9 and s10) demonstrated partial proficiency on both the physics and CT components (scoring 1 point on the physics component and less than 4 points on the CT component). We define high proficiency on a component as scoring the maximum possible points on the rubric for the component. Figure 4 summarizes students' scores.

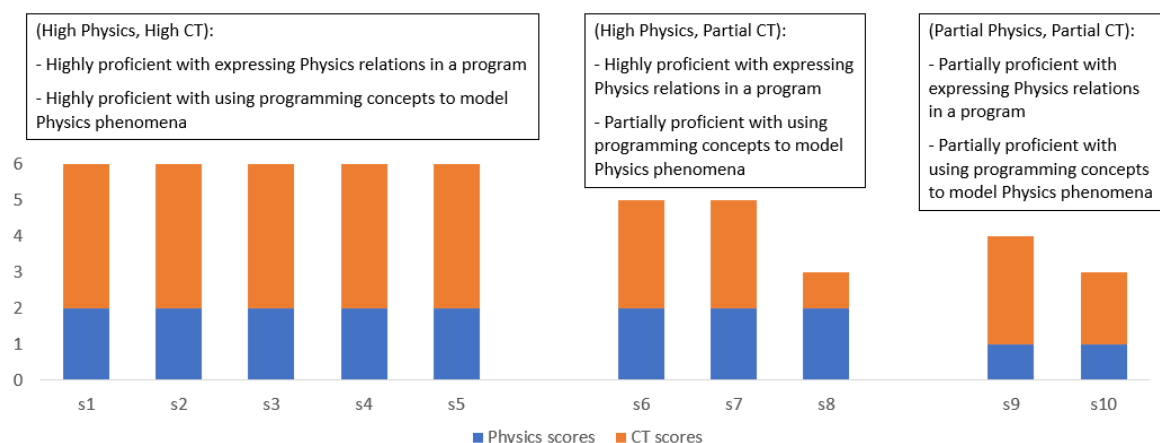


Figure 4. Distribution of students' scores on the Airport task.

Based on students' proficiencies on the two rubric components, we grouped them into three categories – High Physics-High CT, High Physics-Partial CT, Partial Physics-Partial CT. In our small sample, we did not find any student work that we could categorize as Partial Physics-High CT. Next, we discuss example solutions and some of students' programming behaviors for each of the three student categories.

Category 1: High Physics, High CT: Figure 5 illustrates two correct solutions where the only change students made to the given code was modifying the procedure ‘set-Josh-resultant-velocity’ to specify Josh’s new velocity beyond Point B. In the solution to the left, the student correctly specifies Josh’s velocity as the sum of the walkway speed and Josh’s speed in the ‘else’ part of the given conditional block. In the second solution, the student hardcodes the value of the variable ‘Josh’s speed’ to 2.5 instead of a more general expression. In both examples, the students do not modify the other procedure ‘update-position’, thus maintaining correct relations among position, velocity, and time. All parts of the codes are reachable, and the students correctly distinguish between initialization actions that must occur when the green flag is clicked, and actions that must repeat at every simulation step. These programs meet all six criteria across both rubric components.

While all five students in this category finally produced programs that demonstrated high proficiency in physics and CT, we observed that their pathways to reach the final state varied. Two of the students reached the correct solution on their first attempt, requiring only a single test of the modified program. The three other students initially modified Josh’s velocity before Point B (instead of after Point B) and specified a non-zero y-component of Josh’s velocity. However, they were able to rapidly identify the errors and debug their programs.

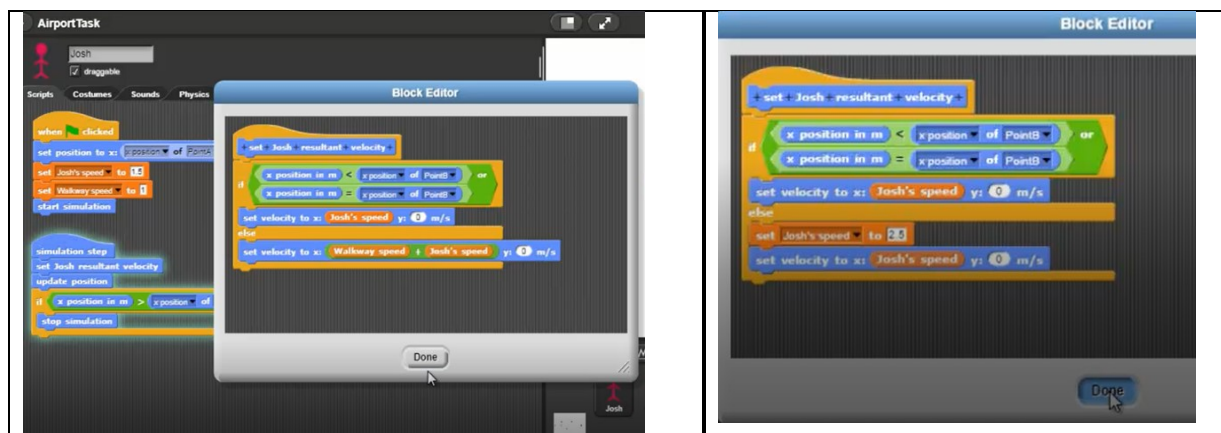


Figure 5. Two examples of correct solutions from the (High Physics, High CT) category.

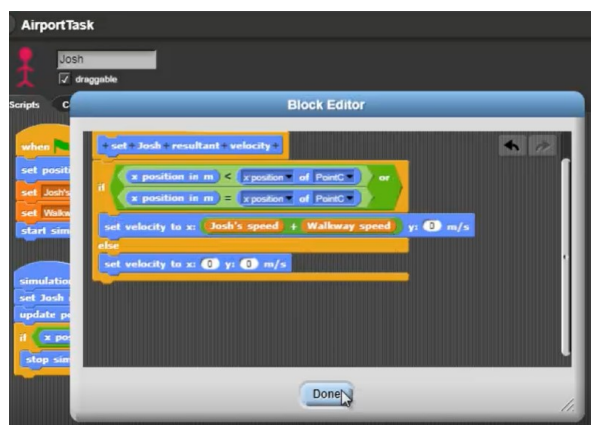


Figure 6. An example solution from the (High Physics, Partial CT) category.

Category 2: High Physics, Partial CT: Figure 6 illustrates one of the three student solutions in this category. The student correctly expresses the relations among velocity, position, and time, and correctly expresses Josh’s resultant velocity as the sum of ‘walkway speed’ and ‘Josh’s speed.’ However, the student incorrectly specifies conditions for updating Josh’s velocity by using all points to the left of Point C instead of only points between points B and C (rubric criterion 5). The incorrect conditional statement is the reason for the program earning less than the maximum score on the programming concepts rubric component.

Based on the videos, we observed that all three students in this category went through multiple iterations of testing and subsequent program modification to reach their final program state, confirming that they made legitimate efforts to reach a correct solution. None of the three students had difficulty creating the physics expression ‘Josh’s speed + Walkway speed’, but they made three general types of programming errors related to the CT constructs of variables, conditionals, and control structures. First, students sometimes assigned the physics

expression to an incorrect variable. Second, students specified incorrect conditions under which the expression applies. Third, students used ‘forever’ loops that do not terminate. These examples illustrate ways that students’ proficiency with a physics and CT concepts are distinct.

Category 3: Partial Physics, Partial CT: Figure 7 illustrates one of the two solutions in this category. The student demonstrated an incomplete understanding of both the physics relations and the programming concepts, scoring 1 and 2 points respectively on the Physics and CT components of the rubric. On the physics component, the student incorrectly expresses Josh’s velocity beyond Point B as the product of time and speed (rubric criterion 1). Also, in the procedure ‘set Josh resultant velocity’, the student has incorrectly set Josh’s velocity beyond Point B to zero (rubric criterion 5), thereby updating Josh’s velocity differently in two places in the code. Moreover, the solution incorrectly contains a ‘forever’ loop inside the simulation step, effectively stopping the execution of other code for all objects (sprites) (rubric criterion 6).

From the videos, we observed that the programming behavior and challenges faced by students in this category were generally similar to that of students in Category 2, except that these students were unable to correct either their Physics related errors or errors from incorrect programming constructs. In fact, the physics-related challenges appeared to be compounded by computational challenges. For example, when a ‘forever’ loop in the simulation step for ‘Josh’ effectively stopped execution of code for other objects (sprites), one student was compelled to modify code for a different sprite (Kate) to model its motion correctly.

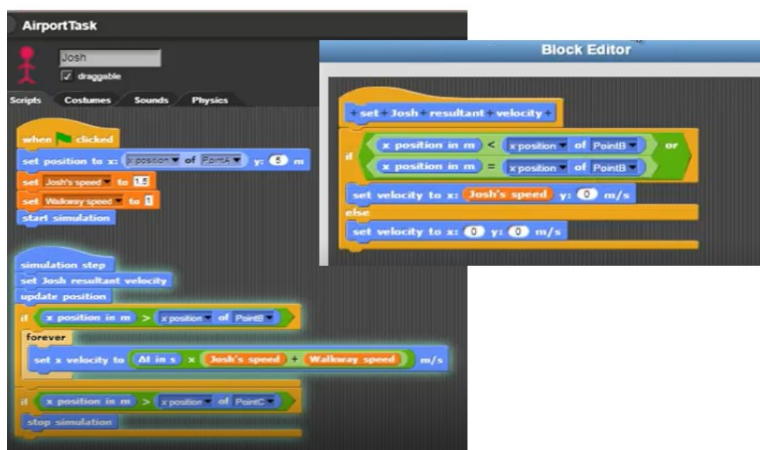


Figure 7. An example solution from the (Partial Physics, Partial CT) category.

Discussion and future work

The recent focus on “CSForAll” (Barnes, 2017) and the policy attention to STEM learning has led to an escalated interest in finding ways to tap into the synergy between CT and science. Making STEM+CT learning successful in precollege settings requires systematically designed assessments for this integrative domain. This paper discusses an approach for designing assessment tasks that target integrated proficiencies across science and CT disciplines, while also differentiating among levels and the nature of proficiencies in the disciplines. The approach has the potential to be generalized to all grade levels and science disciplines. ECD enables us to use a principled approach for assessment development that integrates concepts and practices in the domains while aligning with established education frameworks that integrate content and practice.

Our examination of students’ responses on one such integrated assessment task during a recent pilot study reveal varied physics and CT related challenges that students face while working on such integrative tasks. The ability to identify these challenges can provide valuable information to help teachers guide individual students appropriately (e.g., science disciplinary content versus programming concepts). Our work illustrates the potential value of using such assessments for formative purposes, so that students can achieve synergistic learning of science disciplinary concepts, CT concepts, and computational modeling practices. In order to be useful for formative purposes, assessments must be able to isolate evidence on a specific set of constructs and should not involve additional construct irrelevant activity. Also, varying the task design formats for the same target constructs can help elicit evidence of proficiency at different levels of granularity and provide a more comprehensive assessment of students’ proficiencies.

As future work, we will analyze student responses to integrative assessment tasks from a larger classroom study. We plan to analyze responses to a range of tasks from the kinematics domain and a different physics domain (force) created using the ECD-based approach described above, allowing us to generalize our two-component

rubric framework. Analyzing student work across the two domains will enable us to investigate how students' CT proficiencies change over time and whether they transfer across domains. Further, we will explore ways to apply the rubrics to observable evidence from log files to facilitate automated scoring of these integrative assessments. Manually scoring students' programming artifacts using multi-point rubrics requires going through each students' code and can be labor intensive. While automating the scoring of open ended programming tasks can be extremely challenging, a principled design for focusing on specific constructs in our carefully designed assessment tasks constrains possible student choices and makes automated scoring feasible. Automated scoring will offer opportunities to provide students with carefully designed guidance in real time and provide rapid insights to teachers about their students' proficiencies that can, in turn, inform teachers' instructional decisions.

References

- Basu, S., Biswas, G., Sengupta, P., Dickes, A., Kinnebrew, J. S., & Clark, D. (2016). Identifying middle school students' challenges in computational thinking-based science learning. *Research and Practice in Technology Enhanced Learning*, 11(1), 1-35.
- 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.
- Barnes, T. (2017). CS for all, equity, and responsibility. *SIGCSE Bull.* 49, 2 (May 2017), 18-18. DOI: <https://doi.org/10.1145/3094875.3094882>
- Blikstein, P., & Wilensky, U. (2009). An atom is known by the company it keeps: A constructionist learning environment for materials science using agent-based modeling. *International Journal of Computers for Mathematical Learning*, 14(2), 81-119.
- Broll, B., Volgyesi, P., Sallai, J., and Ledeczki, A. (2016). NetsBlox: a visual language and web-based environment for teaching distributed programming (technical report). <http://netsblox.org/NetsBloxWhitePaper.pdf>.
- Freeman, S., Eddy, S. L., McDonough, M., Smith, M. K., Okoroafor, N., Jordt, H., and Wenderoth, M. P. (2014). Active learning increases student performance in science, engineering, and mathematics. *Proceedings of the National Academy of Sciences* 111, 23, 8410-8415.
- Goode, J., Chapman, G., Margolis, J. (2012). Beyond Curriculum: The Exploring Computer Science Program. *ACM Inroads*. 3(2).
- Hambrusch, S., Hoffmann, C., Korb, J. T., Haugan, M., & Hosking, A. L. (2009). A multidisciplinary approach towards computational thinking for science majors. *ACM SIGCSE Bulletin*, 41(1), 183-187.
- Harris, C. J., Krajcik, J. S., Pellegrino, J. W., & McElhaney, K. W. (2016). Constructing assessment tasks that blend disciplinary core Ideas, crosscutting concepts, and science practices for classroom formative applications. *Menlo Park, CA: SRI International*.
- K-12 Computer Science Framework. (2016). Retrieved from <http://www.k12cs.org>.
- Landau, R. (2006). Computational physics: A better model for physics education? *Computing in science & engineering* 8, 5, 22-30.
- 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. (2012). A framework for K-12 science education: Practices, crosscutting concepts, and core ideas. Washington, DC: National Academies Press.
- NGSS Lead States. (2013). Next Generation Science Standards: For states, by states. Washington, DC: National Academies Press.
- Papert, S. (1991). Situating constructionism. In I. Harel & S. Papert (Eds.), *Constructionism*. Norwood, NJ: Ablex Publishing Corporation.
- Redish, E. F., & Wilson, J. M. (1993). Student programming in the introductory physics course: M.U.P.P.E.T. *American Journal of Physics*, 61, 222-232.
- 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.
- Soloway, E. (1993). Should we teach students to program?. *Communications of the ACM*, 36(10), 21-24.
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM, Viewpoint* 49, 3, 33-35.

Acknowledgments

We thank Nicole Hutchins, Miklos Maroti, Luke Conlin, Kristen P. Blair, Doris Chin, Jill Denner, and our other collaborators from Vanderbilt University and Stanford University for their numerous contributions. This research is supported by NSF grant #1640199.