Yang, J. C. et al. (Eds.) (2018). Proceedings of the 26th International Conference on Computers in Education. Philippines: Asia-Pacific Society for Computers in Education

Studying Synergistic Learning of Physics and Computational Thinking in a Learning by Modeling Environment

Nicole HUTCHINS[a*], Gautam BISWAS[a], Luke CONLIN[b], Mona EMARA[c], Shuchi GROVER[d], Satabdi BASU[e], & Kevin McELHANEY[e]

[a]*Vanderbilt University, USA*
[b]*Salem State University, USA*
[c] *Damanhour University, Egypt*
[d]*Stanford University, USA*
[e]*SRI International, USA*
*nicole.m.hutchins@vanderbilt.edu

Abstract: Synergistic learning of computational thinking (CT) and STEM has proven to effective in helping students develop better understanding of STEM topics, while simultaneously acquiring CT concepts and practices. With the ubiquity of computational devices and tools, advances in technology, and the globalization of product development, it is important for our students to not only develop multi-disciplinary skills acquired through such synergistic learning opportunities, but to also acquire key collaborative learning and problem-solving skills. In this paper, we describe the design and implementation of a collaborative learning-by-modeling environment developed for high school physics classrooms. We develop systematic rubrics and discuss the results of key evaluation schemes to analyze collaborative synergistic learning of physics and CT concepts and practices.

Keywords: Learning-by-modeling, synergistic learning, DSML, collaborative learning

1. Introduction

The need for interdisciplinary STEM skills coupled with collaborative problem-solving abilities has increased significantly in an age driven by technological advancements and globalization. With the increasing reliance on computational modeling and simulation tools in the workplace (Freeman et al., 2014), secondary school students need for developing computational skills is clearly outlined in the United States' Next Generation Science Standards (NGSS, 2013) and the Computer Science (CS) Education (K–12 Computer Science Framework, 2016) frameworks. An approach to ensuring the development of these needed skills is through the synergistic learning of STEM and computational thinking (CT) concepts and practices in classroom environments. Synergistic learning is further amplified, when students collaborate to build and refine computational models, and use these models for problem solving tasks. In particular, computational modeling helps students develop key scientific practices, such as reasoning, explanation, and argumentation.

The driving force behind the integration of CT into STEM domains can be traced to Papert's (1993) pioneering call for inserting constructivist elements through programming into K-12 curricula to enable students to generate and develop powerful ideas. This framework, further invigorated by Wing's (2006) demonstration of the ubiquity of computational thinking concepts and practices across disciplines, clearly lays out the pedagogical benefits of synergistic learning. These include lowering the CS learning threshold by easing the introduction to programming through the use of contextualized representations while also lowering the learning threshold for science concepts by reorganizing them around intuitive computational mechanisms as opposed to equation-based formulations (Sengupta et al., 2013). These benefits have been actualized through studies run with computer-based learning environments (e.g., CTSiM (Basu et al., 2017) and Netlogo (Weintrop et al., 2016)), further motivating the need for developing sustainable, synergistic learning environments that can proliferate this form of learning across multiple STEM domains.

This has led to our designing and developing the Collaborative, Computational STEM (C2STEM) learning environment (Hutchins et al., 2018). C2STEM is based on a novel paradigm that combines visual programming (Kelleher & Pausch, 2005) with domain specific modeling languages (DSMLs) that promote the learning of discipline-specific and CT concepts and practices through computational model building and problem-solving exercises. DSMLs allow students to express and develop computational solutions at a level of abstraction that is aligned with the STEM constructs they are required to learn, while developing models that are self-documenting (Hasan & Biswas, 2017). This lowers the barriers for students to experiment with different modeling and problem-solving ideas, analyze the models they develop, and check for correctness using their intuitions of system behavior or information they learn from the provided instructional and assessment opportunities. In addition, we have adopted an evidence-centered design (ECD) approach (Mislevy & Haertel, 2006) based on established standards for task and assessment (both formative and summative) development, making the system amenable for classroom adoption. We use a design-based research (DBR; Cobb et al., 2003) approach to system development by continually experimenting with the affordances and scaffolds provided in the system, and refining them through iterative development to meet students' needs and address their learning difficulties.

This paper discusses a collaborative, learning-by-modeling study that we conducted with a small group of high school students working on a forces curricular unit. We analyzed the verbal protocols and the progression of models that the students generated to identify synergistic learning instances and their outcomes. The goal was to answer the following research question: *Do students that work collaboratively in C2STEM adopt synergistic learning mechanisms that help them gain a better understanding of relevant STEM and CT concepts and practices*? To address this question in greater depth, we address two additional questions: (1) *What is the framework needed to support collaborative learning of STEM concepts and practices in C2STEM*? and (2) *What components of C2STEM provide opportunities for synergistic STEM and CT learning*? To address these questions, we present a case study from an experiment we ran with 25 high school students who collaborated on their model building tasks in groups of 2 or 3 in a classroom environment.

2. Background

2.1 Learning by Modeling as a Framework for Synergistic Learning

C2STEM supports synergistic STEM and CT learning by getting students to construct computational models of scientific processes. We hypothesize that the affordances provided in constructing a step-by-step simulation model of the motion of objects (as opposed to traditional equation-based modeling) and then visualizing their behavior evolution using animations and plots provides students with a deep understanding of the primary kinematics concepts, e.g., the relations between acceleration, velocity, and position of an object. This is further facilitated by the use of DSMLs that keep the focus on domain concepts (Hasan & Biswas, 2017), and supported by visual, block-structured constructs that relieve students from the burden of learning the syntax of programming languages, while keeping the focus on important CT concepts and practices, such as algorithmic structures, use of variables, their initialization and update functions, and debugging and refinement to build correct and complete models (Grover & Basu, 2017). In addition, links between computational model building and real world phenomena can further improve understanding of core STEM concepts and practices and facilitate thinking in different ways about other disciplines (Angeli et al., 2016). The use of computational constructs in building models of familiar physical phenomena provides students with a framework to learn CT concepts and practices in context, thus making the interpretation and use of computational processes more grounded as opposed to being studied as arbitrary logical structures. On the other hand, having to represent relations between physics concepts in explicit computational structures, and then executing these structures (models) to observe the generated behaviors, provides students with immediate feedback, and the ability to reason about the impact of the physics laws on the behavior of the modeled objects. This provides the framework for synergistic learning, and past studies have demonstrated that it results in increasing learning gains in the physics and CT domains (Basu, et al., 2016; 2018; Hutchins et al., 2018).

C2STEM's learning by modeling approach extends from the successes of previous synergistic learning environments, such as CTSiM (Sengupta et al., 2013), ViMap (Sengupta et al., 2015), and CT-STEM (Jona et al., 2014). As discussed above, the affordances attributed to learning-by-modeling approaches have significantly enhanced synergistic learning of STEM and CT content. Not only does the open-ended nature of constructing, debugging, and solving authentic problems coincide with the notion of generating powerful ideas (Papert, 1993) and putting them to practice (Wilensky & Resnick, 1999), it allows students to take ownership of their learning, utilizing tools and methods that scaffold their understanding, analyzing, and problem solving.

2.2 Collaborative Problem-Solving

Collaborative learning research continues to advance our understanding of the cognitive and metacognitive processes that students articulate when working in groups to address problem solving tasks (Emara et al., 2017). Roschelle and Teasley (1995) defined collaboration as "a coordinated, synchronous activity that is a result of a continuous attempt to construct and maintain a shared conception of a problem." Dillenbourg (1999) further characterized collaboration by symmetry in actions, knowledge, skills, and status among peers. Research has shown that successful completion of collaborative learning tasks requires the development of a shared understanding among group members (Larkin, 2006), as well as interdependent skills such as contributing and encouraging contribution of ideas, monitoring and reflection of group progress, and providing constructive feedback to group members through argumentation and explanation (Garrison & Akyol, 2013). In addition, previous studies have examined conceptual changes in group knowledge based on collaborative conversational interactions (Rochelle, 1992) and co-creation of web-based science inquiry models by students who were not physically co-located (Gobert et al., 2007).

C2STEM offers a computer-supported, collaborative learning environment, where students can co-construct knowledge by building their computational models of STEM topics in a shared workspace (Hutchins et al., 2018). Moreover, working together face-to-face provides them opportunities to discuss the model construction process, generate explanations for other group members based on the group's computational models, and develop arguments that support or challenge model constructs proposed by their partners (Sins, Savelsbergh, & van Joolingen, 2005). As a means of supporting co-construction of knowledge through model building and problem-solving activities, our aim in designing the learning environment and the accompanying curriculum was to promote the necessary knowledge development and debugging skills through communication and discussion opportunities. To our knowledge, research has not yet analyzed actions and discourse data on how collaborative learning-by-modeling efforts promote synergistic learning of STEM and CT.

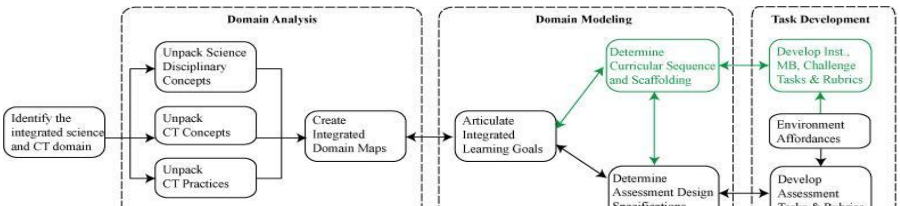

Figure 1. Overall Curriculum and Assessment Design Process.

3. Design & Implementation of C2STEM

3.1 Curriculum Development

Our research team has adopted ECD as a means for supporting the development of learning and assessment tasks. ECD promotes coherence in the design of assessment tasks and rubrics, and the interpretation of students' performance by explicitly linking claims about student learning, evidence from student work products, and design features of tasks that elicit the desired evidence. As an extension to our detailed assessment design process, Figure 1 outlines the design process utilized for developing the collaborative, learning tasks and associated pre-post questions. The main addition of this paper is included in green in the figure. Components of this model will be discussed in the following subsections. Student performances in the pilot study were evaluated as part of our DBR process in order to analyze the application of our integrated learning goals and accompanying tasks. Curricular adjustments were made based on this process (for instance, in a pilot study, we saw little improvement in students' abilities to interpret velocity-time and position-time graphs, so we implemented additional instruction on the use of the graphing tool).

Curriculum implementation is initiated by a thorough analysis of the curriculum's targeted concepts and practices, i.e., learning of relevant concepts in motion generated by forces along with the relevant CT concepts and practices that support computational modeling and analysis of motion due to forces. Through domain analysis, we unpacked disciplinary concepts utilizing the key physics and computing standards discussed earlier, and sought input from the classroom physics teacher on how to best develop the instructional and modeling units for classroom instruction and learning. Example force concepts can be found in Table 1. CT concepts and practices included: initializing and updating variables, operators and expressions, and control structures (event handlers, conditionals, iterations) as concepts, and CT practices that included the use of algorithmic structures to develop computational models and debugging of models to ensure their correctness and accuracy. The unpacking of these concepts led to the development of integrated forces and CT learning goals that we addressed in the curriculum (Basu et al., 2018).

Table 1

Example Concepts Developed for Each Domain

Phy.	Concepts
1	Force = mass * acceleration; Greater the net force, greater the acceleration
2	Resultant horizontal (or vertical) force = Applied force – Frictional force (or weight) *[excluding frictional forces (e.g. air resistance) in the y-direction]*
3	An object at rest will move only if applied force is greater than max. force of static friction

Utilizing the learning goals established in the domain analysis process, our team developed curricular tasks targeted towards each of our stated learning goals to ensure that the relevant curricular concepts and practices in physics and CT were covered by the set of developed tasks. In addition, the mapping of learning objectives to curricular tasks provided the assurance that key learning objectives were applied at varying levels of complexity with the goal of supporting learning gain opportunities.

We developed assessment tasks to measure proficiencies in implemented force and CT constructs using an ECD approach (Basu et al., 2018). Given in-class time constraints, embedded assessments were integrated in the form of task submission questions for each learning task. This allowed for the research team to monitor for potential misunderstandings students had in physics and CT concepts. Summative assessment was implemented using a pretest and posttest, with questions designed to target key force and CT concepts and practices. For instance, four of the 11 assessment questions were designed to evaluate student understanding of "Force = mass × acceleration (greater the net force, greater the acceleration)."

3.2 System Design and Implementation

The current version of our system is designed for high school physics classrooms. Due to the limited availability of K-12 computer science teachers, we assume that classroom participants have limited exposure to programming in their formal curriculum. As such, it is important to note that the introduction of this synergistic learning environment into a physics classroom requires the assurance that students will receive all of the necessary physics instruction, without consuming significant chunks of classroom time on training in block-based programming. However, students who have not been exposed to programming and CT, building computational models can be a non-intuitive and difficult task (Grover & Basu, 2017). Therefore, a seamless, well-designed introduction to CT concepts and practices is an important consideration to avoid additional barriers that may impede both domain and CT learning.

The C2STEM learning environment is web-based and runs off a cloud-based server. Therefore, students have ubiquitous access at school and out of school to work on their learning and problem-solving tasks. The computational modeling representation uses custom physics

DSML blocks developed on top of NetsBlox (Broll et al., 2016), a block-based extension of Snap! (http://snap.berkeley.edu/). Prior to conducting the collaborative study on the force module, we designed and implemented new DSML blocks specific to the modeling motion of objects in the presence of forces (e.g. "set net force in *x*-direction to (*value*) *N*" and "*set x-acceleration to net force in x-direction / mass of object*"). The rest of the physics computational constructs were the ones developed for the kinematics modules that preceded the force module (Hutchins et al., 2018).

Our learning-by-modeling curriculum is designed to support the classroom Physics instructor through four levels of curricular tasks: (1) instructional, (2) model building, (3) embedded assessment, and (4) challenge (Hutchins et al., 2018). From the highly scaffolded *Instructional tasks* to the more comprehensive *Challenge tasks*, each curricular task involves increasing application of more complex STEM and CT concepts and practices to support student learning through model building and problem solving. For instance, in the forces unit, students first complete *Instructional tasks* that introduce vertical forces (due to gravity). This is followed by a *Model Building task* in which students have to determine the maximum acceleration that a package being lifted by a drone can achieve given the maximum lift force of the drone. This task allows for an introduction to global (gravitational force) and local variables (e.g., the resultant lift force acting on the package). The next component of the forces unit involves an introduction to horizontal forces via *Instructional tasks*. This is again followed by a *Model Building task* in which the students have to construct a push-buggy model to move a package that is initially at rest at a given constant velocity. The two phases of the model behavior: (1) accelerating from rest, and (2) moving at a constant velocity is implemented using conditional logic to distinguish the different phases. Embedded formative assessment tasks are interwoven between *Instructional* and *Model Building tasks* to help students assess their learning of key STEM and CT concepts and practices required for that module. Finally, in the Forces *Challenge task* (see Figure 2), students are required to use the push-buggy and a drone to move a package that is too heavy for the push-buggy to move on its own. Such complex problems are best solved by decomposing the problem into parts (a CT practice), and constructing and testing each part, before composing and testing the complete model. The composition process may require the use of complex conditional constructs and the careful choice of local and global variables to make the construction and debugging processes easier to handle.

4. Classroom Study

4.1 Participants and Procedure

We conducted a two-month-long study with 26 10th grade students who worked in groups of 2 or 3 on C2STEM as part of their classroom activities. The research team met with participants one school-day a week over a two-month period. Our primary aim in this study was to determine how students collaborated in the C2STEM environment.

Students completed one 45-minute CT training unit and four physics modules: three in Kinematics: 1D motion (with acceleration), 2D motion with constant velocity, and 2D motion with gravitational forces, and one in mechanics, i.e., an introductory unit on 1D Force. All work was completed collaboratively, either in pairs or in triads. Class lectures were led by two members of the research team that included a former high school teacher. In the classroom, teams primarily worked on model building tasks. Homework assignments consisted of instructional tasks. In the classroom each group worked on one laptop while interacting face to face with each other. Outside of the class, teams collaborated via a Google Groups forum. A member of the research team also monitored the forum to answer student questions. On the last day, teams presented the final solution to the challenge problem from any of the previously described modules, and the process they employed to solve their problem. Students completed the force pretest individually prior to the CT training unit and the posttest individually following their final presentations.

4.2 Data Sources and Analysis Plans

The primary source of assessment analysis included a paper-based pre-post assessment targeting Force concepts described in section 3.1. Rubrics for each task were developed as the questions were developed. In addition, groups were tasked with submitting multiple choice and short answer questions related to the simulation models they developed for each task.

As a means of analyzing key synergistic learning events, we recorded all group work using the OBS™ screen-capture software. In addition, our modeling environment has a "replay" mode, which allowed researchers to view how student work on the system evolved over time. We examined student recordings to characterize their approach to model building, the challenges they faced, and the discourse mechanisms they employed as they worked with each other. For each task, we noted key actions and conversations that highlighted their synergistic learning events. As a means of coordinating the identification of synergistic learning activities, a rubric was created to identify different levels of synergistic learning applications (i.e., low, medium, and high). Table 2 illustrates example student work linked to the three levels, and provides a basis for evaluating how students adopt and move between these levels as they go about their model building tasks. In addition, as an example, if we find students struggling in applications with level 3 CT and level 1 physics, it would alert our team to adjust the physics instruction to help students get beyond their preliminary understanding of these concepts, and apply them in building their computational model.

Table 2

Identifying Synergistic Learning during the Video-Coding Process

Level	Physics	Example	CT	Example
1	Low	Using Physics DSML	Low	Using programming blocks
2	Medium	Describing object movements	Medium	Initializing, updating variables
3	High	Describing causal relations	High	Setting up conditionals to model behavior phases

5. Results and Analysis

5.1 Do students that work collaboratively on a learning-by-modeling, STEM and CT task improve in their understanding of the relevant STEM domain?

As mentioned previously, students completed four curriculum tasks collaboratively (three instructional tasks and a challenge task). The force pretest and posttest included 10 questions worth 21 points targeted to evaluate domain knowledge based on established learning goals. Participants earned an average score (standard deviation) of 4.56(2.02) on the pretest, increasing to a 6.37(3.25) on the posttest. A t-test ($t(50) = -2.406, p = 0.02$) indicated students had significant learning gains. In the posttest, students improved in drawing free body diagrams, suggesting improvement in their conceptual understanding of the forces at play in a given problem. Students continued to struggle in their understanding of "non-zero resultant force implies non-zero acceleration, and zero acceleration implies zero net or resultant force" as well as their ability to use equations (e.g., resultant force = applied force – frictional force). Given the low performance on the posttest compared to total possible points, we believe additional instructional and model building tasks with embedded check-ins (formative assessment) could improve students learning over time (Hutchins et al., 2018; Basu et al., 2018).

5.2 Case Study: Collaboration

In this paper, we limit our case study analysis to one model building task, i.e., the Force Challenge task (Figure 2), and compare students' final models with their performance on questions 6 and 7 of our pretest and posttest. Question 6 is designed to target students' abilities in drawing free-body diagrams of the forces acting in the system. Question 7 evaluates students' knowledge on "maximum static frictional force equals the coefficient of static friction multiplied by normal force" and "an object at rest will move only if applied force is greater than maximum force of static friction." The ECD process identified the challenge problem as addressing these constructs.

The group for this case study consisted of two male students who listed no previous classroom experience in Physics. One student had previous experience with the C2STEM environment.

On the pretest, Student 1 received a score of 4 and Student 2 received a score of 6. For Question 6, Student 1 received 1 out of 2 points, while Student 2 received 1.5 out of 2. Both students failed to illustrate normal force and Student 1 did not take gravity into account. For Question 7, both students received 0 out of 4 in the pretest.

5.2.1 Collaborative Learning-By-Modeling

Prior to completing the forces challenge task, the group received feedback from a researcher on their instructional task submissions (including feedback on the creation of variables not available in the physics DSML, the affordances of utilizing operators to set and change physics variables, and the requirement of programming all objects in the model). For this analysis, we analyze the recorded video and audio of the group. For the challenge problem, the group elected to begin by completing their needed calculations on paper. Their initial collaborative activity prior to beginning their simulation had a level 3 label for Physics and a level 0 label for CT. To begin the task, Student 1 articulated the problem by stating, “you need to lower the normal force until 500 Newtons is enough to overcome static friction.” This articulation identified the first instance in which the group recognized normal force as a required concept for model building. Continuing on, another group sitting at the table asked Student 1 how to solve for static friction. While Student 1 corrected the other group in their use of mass of the object instead of the object’s weight, Student 2 calculated the value needed. Student 1 followed the conversation noting that “you need to keep lowering the normal force until that equals 500,” which would not be sufficient to move the object (see concept #3, Table 1).

After their calculation on paper, the group elected to hardcode values for all programmed properties (other than net force – see Figure 3) to build their model. These activities received the label $Physics\ level = 2, CT\ level = 2$. After some time, a student from another group interjected “we have to make static friction like 499.9.” Student 1 asked why, to which the other student replied “so if it’s pushing the package at 500 and since static friction is 500, it’s not going to move anything.” A student some distance away asked the class, “What’s the 499 from?” Student 1 articulated that “we have to lower the normal force enough so that the static friction goes from 588 to 499, because if the [buggy] is pushing at 500 and the static friction is at 500 the forces balance out and the box doesn’t move.”

An analysis of the replay mode for this project indicate that the group utilized the same start process as the first instructional task in which they initialized needed physics-DSML variables under the green flag for the push-buggy object instead of organizing variables first. The students hardcoded the net x-force and did not create a static friction variable. Then they began developing their “simulation step” code for the push-buggy. The students demonstrated the CT practice of code reuse by duplicating the push-buggy initialization and simulation code for the package and the drone. This was an improvement on their work in the instructional task. They also showed other improvements in variable initialization by using a global lift-force variable. The students then completed the code to describe the movement of the box, and replicated this for the drone. They disconnected their previous initialization code from the green flag so it would not run when the flag was clicked, but they kept the code on the stage.

Separate from the analyses of learning gains and synergistic learning examples, it is important to note the role of the two students in the group. As can be seen, Student 1 played a vocal role in the development of the solution and assisted other students in the class. We also recognized that proper articulation of the problem helped both students better understand their model building task, and the next steps in the implementation. This represents a clear example of developing shared understanding of the problem in collaborative problem solving (Larkin, 2006). For instance, in the above example, Student one stated “you need to lower the normal force until 500 Newtons is enough to overcome static friction.” Over the 30 minutes of video analyzed for this paper, Student 1 articulated the problem 4 times, while Student 2 articulated the problem 1 time. In addition, there were three instances when Student 1 repeated the problem articulation to students in other groups.

5.2.2 The Final Code and Assessment Results

The final submission by this group is shown in Figure 3. The values hard-coded in the simulation are different from those found on the group's final presentation on their program. We presume this is due to a failure to save an updated version of their project. Given the CT rubric outlined in Table 2, it is important to note that our instructions did not specify stopping conditions or a location in which the package was to be placed. We did not explicitly assess the requirements for Criterion #2. However, as can be seen in the code, issues did exist in terms of the appropriateness and correctness of variables. In addition, the students implemented a global "Drone Lift" variable (in orange), but set it in both the package's and drone's code. Fortunately, the value was set to the same number for both. The use of one object's property on another objects behavior indicates a preliminary understanding of local and global variables, while also showing the physics relationship required.

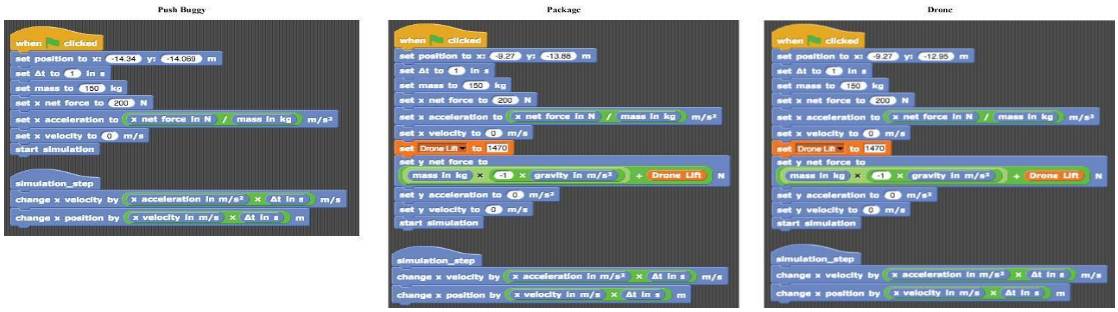

Figure 3. Group One Solution to Force MB Task

Finally, for the posttest regarding questions 6 and 7. Both students received a perfect score for Question 6, indicating a learning gain regarding normal force. For Question 7, however, Student 1 scored a 4 out of 4, but Student 2 remained at a 0 out of 4. Upon closer examination, Student 2 was unable to calculate the static frictional force correctly. Given the control of the problem-solving process by Student 1, it would be worthwhile to explore ways of supporting group dynamics that foster learning for all members. One possibility to explore could be to encourage students to actively switch leadership positions over the course of a challenge problem.

6. Discussion

6.1 What is the framework needed to support students in learning STEM concepts and practices through collaborative, computational model building?

This research question implies generalizability in a design process for applications of more than one STEM domain with CT. It can be seen through this work that a detailed ECD process with accompanying domain and CT unpacking and the development of integrated learning objectives can be utilized for an effective implementation of synergistic STEM and CT learning tasks. In addition, a detailed outline of learning goals for both the STEM domain and CT supports the application of the STEM and CT evaluation tool (Table 2). In this study, the "Unpacking of Disciplinary Concepts" (Figure 1) focused on unpacking concepts needed to teach a high school physics unit on forces. In our previous work (Hutchins et al., 2018, Basu et al., 2018), this process has been successful for physics units on kinematics. Furthermore, a scaffolded implementation designed to first target specific physics concepts (e.g., a vertical force task and a horizontal force task) and needed CT practices (e.g., creation, initialization and updating of variables) with relevant feedback promoted improvements in synergistic learning. Our initial results show that this results in students learning physics concepts through model building tasks.

6.2 What components of the learning-by-modeling approach provide synergistic STEM and CT learning opportunities?

As a conclusion to our analysis, we review a few key elements of synergistic learning provided by our learning-by-modeling environment. The first being the use of a physics DSML to program the model. During each of the learning tasks, students were required to utilize physics-domain blocks as they developed their code, allowing for self-documentation of their physics problems. Given our preliminary rubric for identifying synergistic learning, as long as students were programming with the physics blocks in the block-based programming environment, there was a minimum of level 1 Physics and level 1 CT activity by the students. Over time, we have seen this level of achievement as a benefit to synergistic learning (Hutchins et al., 2018).

The use of the modeling environment itself proved to be a useful component in support of synergistic learning. For example, when the package did not move after lowering the static frictional force to 500, students had to find out why their model did not work. If students were to have only submitted their paper calculations for this task, the lack of immediate feedback would not have likely corrected this misunderstanding before submission. This activity showcases a useful application of physics with computational modeling and the CT practice of debugging to resolve a physics misunderstanding. In addition, the separation of the initialization of variables under the green flag block and the updating of the variables under the simulation step flag may have played a role in the understanding of variables needed for x-directional and y-directional movement.

Finally, this challenge problem required the student to program three objects to work together to resolve a problem. Their physics knowledge was challenged through the need to calculate forces appropriately for each object given their interactions with other objects. Their computational knowledge was challenged in that all three objects needed to be programmed correctly for the simulation to generate correct results. However, as mentioned in section 5.2.1, additional CT instruction may help to better introduce the concept of encapsulation in the context of object-oriented programming.

7. Conclusions and Future Work

This paper outlines the design process, implementation, and evaluation of a learning-by-modeling curriculum aimed at the synergistic learning of forces and CT. In addition, a new rubric for evaluating levels of synergistic learning was implemented to highlight synergistic learning opportunities for students utilizing similar learning environments. Our results indicate that: (1) a systematic design approach to the developing curriculum tasks (learning and assessment) is beneficial in addressing key domain concepts and practices to be addressed by the curriculum; (2) students working collaboratively on learning-by-modeling tasks are successful in co-constructing models of physics phenomena; (3) identification of levels of synergistic learning applications is beneficial in the development and evaluation of instructional material and student progress; and (4) the control of the problem-solving process by one group member in a learning-by-modeling environment may affect individual learning gains.

In future work, we will continue to identify synergistic learning opportunities to aid the learning processes in the C2STEM environment. In addition, future plans include scaled up empirical studies in varied contexts with diverse learners so we can build more inclusive scaffolds and generalize the curriculum. Furthermore, we will explore ways this identification process can happen through log files as this may provide for useful personalization of learning tools and the creation of useful student data for classroom teachers.

Acknowledgements

We would like to thank our additional collaborators at Vanderbilt University, Stanford University, and SRI International. NSF DRL STEM +C partnership program grant # 1640199.

References

Angeli, C., Voogt, J., Fluck, A., Webb, M., Cox, M., Malyn-Smith, J., & Zagami, J. (2016). A K-6 computational thinking curriculum framework: Implications for teacher knowledge. *Journal of Educational Technology & Society, 19(3).*

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., McElhaney, K., Grover, S., Harris, C., & Biswas, G. (2018) A Principled Approach to Designing Assessments That Integrate Science and Computational Thinking. *Proceedings of ICLS'18 (*pp.384–391*)*. London, England: ISLS.

Broll, B., Volgyesi, P., Sallai, J., and Ledeczi, A. (2016). NetsBlox: a visual language and web-based environment for teaching distributed programming (technical report). http://netsblox.org/NetsBloxWhitePaper.pdf.

Cobb, P., Confrey, J., diSessa, A., Lehrer, R., & Schauble, L. (2003). Design experiments in educational research. *Educational Researcher, 32(1)*, 9–13.

Dillenbourg, P. (ed.). (1999). Collaborative learning: Cognitive and computational approaches, *Advances in Learning and Instruction Series*. Elsevier Science, Inc, New York, NY.

Emara, M., Tscholl, M., Dong, Y., & Biswas, G. (2017). Analyzing Students' Collaborative Regulation Behaviors in a Classroom-Integrated Open Ended Learning Environment. *Proceedings of CSCL'17* (pp. 319–326). Philadelphia, PA: ISLS.

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.

Garrison, D. R., and Akyol, Z. (2013). Toward the development of a metacognition construct for communities of inquiry. *The Internet and Higher Education*, *17,* 84–89.

Gobert, J., Slotta, J., Clarke, J., Dede, C., Gijlers, H., Saab, N., & Koedinger, K. (2007). Fostering peer collaboration with technology *Proceedings of CSCL'07* (pp. 23–27). ISLS.

Grover, S. & Basu, S. (2017). Measuring Student Learning in Introductory Block-Based Programming: Examining Misconceptions of Loops, Variables, and Boolean Logic. *Proceedings of SIGCSE Technical Symposium on Computer Science Education* (pp. 267–272). New York, NY: ACM.

Hasan, A., & Biswas, G. (2017). Domain specific modeling language design to support synergistic learning of STEM and computational thinking. Kong, S. C., Sheldon, J., & Li, K. Y. (Eds.). *Proceedings of CTE'17* (pp.28–33). Hong Kong: The Education University of Hong Kong.

Hutchins, N., Biswas, G., Maroti, M., Ledeczi, A., & Broll, B. (2018). A design-based approach to a classroom-centered OELE. *Proceedings of AIED'18* (pp.155–159). London, England: Springer.

Jona, K., Wilensky, U., Trouille, L., Horn, M. S., Orton, K., Weintrop, D., & Beheshti, E. (2014). Embedding computational thinking in science, technology, engineering, and math (CT-STEM). *In Future Directions in Computer Science Education Summit Meeting*, Orlando, FL.

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

Kelleher, C., and Pausch, R. (2005). Lowering the barriers to programming: A taxonomy of programming environments and languages for novice programmers. *ACM Computing Surveys, 37*(2), 83–137.

Larkin, S. (2006). Collaborative group work and individual development of metacognition in the early years. *Research in Science Education*, *36*(1–2), 7–27.

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. Washington, DC: National Academies Press.

Papert, S. (1993). *Mindstorms: Children, computers, and powerful ideas*. Basic Books, New York, NY.

Roschelle, J. (1992). Learning by collaborating: Convergent conceptual change. *Journal of the Learning Sciences, 2*(3), 235–276.

Roschelle, J., and Teasley, S. D. (1995) The construction of shared knowledge in collaborative problem solving. *Computer Supported Collaborative Learning*, 69–97.

Sengupta, P., Dickes, A., Farris, A. V., Karan, A., Martin, D., & Wright, M. (2015). Programming in K-12 science classrooms. *Communications of the ACM, 58*(11), 33–35.

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.

Sins, P, Savelsbergh, E., & van Joolingen, W. (2005) The Difficult Process of Scientific Modelling: An analysis of novices' reasoning during computer-based modelling. *International Journal of Science Education, 27*(14), 1695–1721

Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., Wilensky, U. (2016). Defining Computational Thinking for Mathematics and Science Classrooms. *Journal of Science Education and Technology*, *25*(1), 127–147.

Wilensky, U., & Resnick, M. (1999). Thinking in Levels: A Dynamic Systems Perspective to Making Sense of the World. *Journal of Science Education and Technology, 8*(1).

Wing, J. (2006) Computational thinking. *Commun. ACM, 49*(3), 33–36.