Ogata, H. et al. (Eds.) (2015). Proceedings of the 23rd International Conference on Computers in Education. China: Asia-Pacific Society for Computers in Education

Relations between modeling behavior and learning in a Computational Thinking based science learning environment

Satabdi BASU, Gautam BISWAS*, John KINNEBREW & Tazrian RAFI
Department of EECS/ISIS, Vanderbilt University, USA
*gautam.biswas@vanderbilt.edu

Abstract: Computational Thinking (CT) is becoming a core element of current K-12 science curricula. In spite of the known synergies between CT and science education, few attempts have been made to leverage this synergy, especially at the middle school level. In previous work with the Computational Thinking for Simulation and Modeling (CTSiM) environment (Basu, et al., 2013; 2014; Sengupta, et al., 2013) we have demonstrated strong synergies between learning science content and CT, with students showing strong learning gains in kinematics, ecology, and CT concepts. However, when working in CTSiM, students also faced a number of difficulties in understanding science concepts and CT constructs, and had to be scaffolded 1-1 by the classroom teacher or the researchers conducting the study (Basu, et al., in review). To better scaffold their learning, we developed a set of hypertext resources and formative assessment quizzes in the system. This paper reports a teacher-led, multi-domain classroom study conducted with 5th grade students using CTSiM. Our results, based on pre- to post-test gains, and the accuracy of the students' CTSiM models, demonstrate significant and synergistic learning of science concepts and CT skills in Kinematics and Ecology domains, along with transfer of CT skills across domains.

Keywords: Computational Thinking, science learning, agent-based modeling and simulation, visual programming, CT assessments

1. Introduction

Computational Thinking (CT) (Grover & Pea 2013; Wing 2011) has been steadily gaining importance as a vital ingredient for STEM education (NRC, 2011). It is becoming a key component of various national science education frameworks (Guzdial, 1995), which place a lot of emphasis on exploiting the synergies between CT and STEM for classroom instruction. However, few systems or curricula that integrate CT skills with curricular science learning have actually been developed for use in K-12 classrooms (Grover & Pea, 2013). Towards this end, we have been developing CTSiM (Computational Thinking using Simulation and Modeling) (Basu, et al., 2013; Sengupta, et al., 2013) – a CT based science learning environment for middle school students that helps students learn science by constructing computational (i.e., simulation) models of science phenomena.

Our initial studies with CTSiM, used by a science teacher in a 6th grade science classrooms have shown promising results in synergistic learning of science and CT concepts (Basu, et al., 2013; 2014). However, students face a number of difficulties in understanding and applying science domain knowledge and CT constructs to building and debugging their computational models. In studies we have run to date, the classroom teacher or members of our research team have provided additional support to help students overcome their difficulties. To help students tackle these difficulties on their own, we have added resources and formative assessments to the CTSiM environment that help students learn and test their understanding of relevant science and CT concepts. These additions to the system represent an initial step in our goal for providing individualized scaffolding to help students develop more self-regulation skills and apply them to developing correct models of the science phenomena. Though researchers have emphasized the need for scaffolding in CT-based environments, such functionalities are virtually non-existent in the existing CT-based systems (Basu, et al., 2013).

This paper discusses results from a recent classroom study with about 26 5th grade students. Students were assessed on their science and CT knowledge before and after they completed all of the

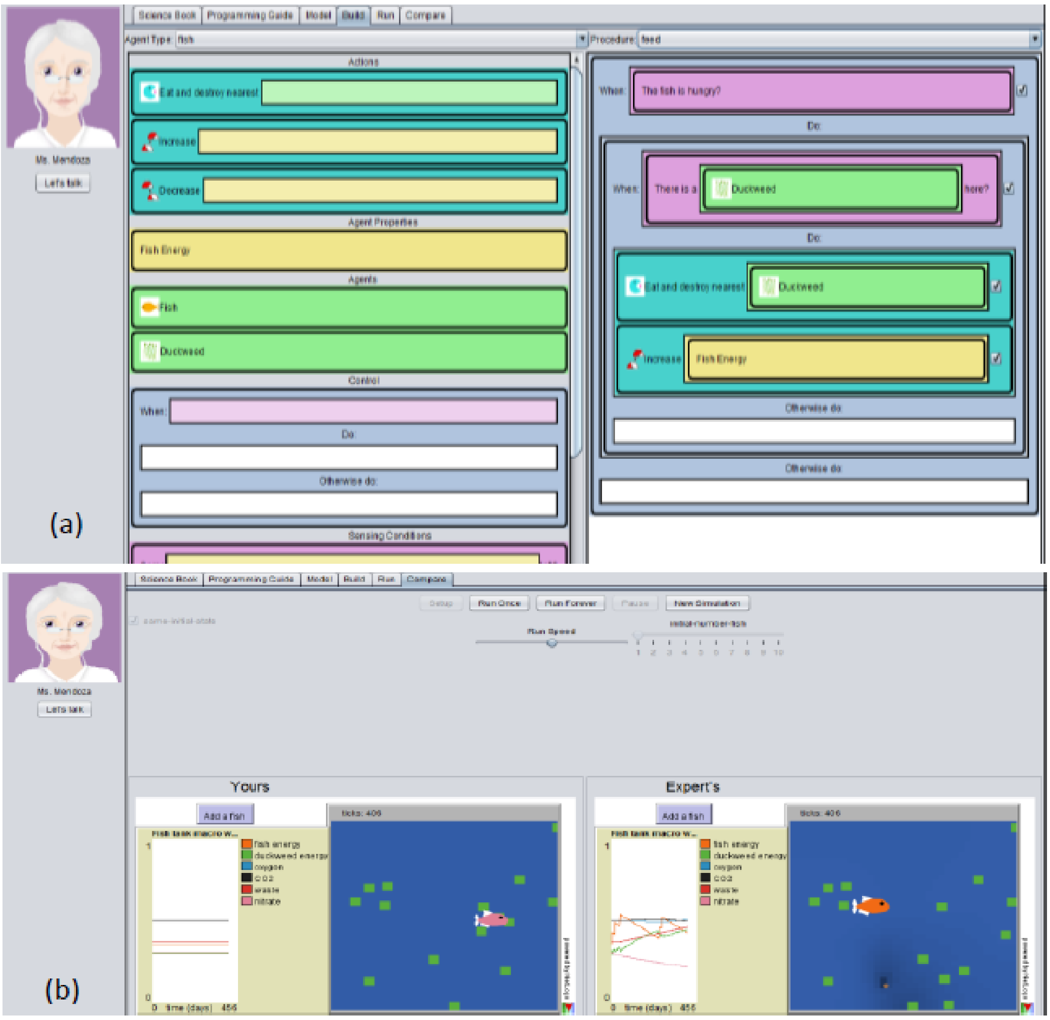

Figure 1(a): The Construction world; (b) The Envisionment world in a fish-tank unit

units for each topic. Students' final models and the evolution of the models within and across units were also evaluated by comparing the models against corresponding correct or 'expert' models for each unit. In the analysis presented in this paper, we investigate the relations between students' pre-post learning gains, information acquisition behaviors, and their ability to build correct models. Our results show strong correlations between the ability to build models and pre-post test gains

2. The CTSiM Environment and Learning Units

Principles that govern the design and implementation of the CTSiM system have been discussed in Basu, et al. (2013) and Sengupta, et al. (2013). The overall focus has been very much on learning by modeling using an agent-based framework. In addition, by developing a visual block-structured domain-specific computational modeling language, students are able to construct, test, and evaluate their evolving models, learning and interpreting new knowledge as they work in this environment.

In CTSiM, students first construct a conceptual model by describing the model structure in terms of agents, their properties, behaviors, and interactions with other agents and the environment. The conceptual model provides the basis for constructing the computational model in the 'Construction world' (see Figure 1(a)). To construct their computational models, students select from a library of visual primitives provided on the left pane and spatially arrange them on the right pane. The available primitives include domain-specific (e.g., 'energy') and domain-general (e.g., conditionals and loops) constructs. Students can observe the behavior of their models by executing them as agent-based simulation (Wilensky & Reisman, 2006), and trace the execution of their models. Students can also verify the correctness of their models by a side-by-side comparison of their models' simulations with an 'expert' simulation in the 'Envisionment world' (see Figure 1(b)). Identifying differences between the simulations helps students identify and correct errors in their models.

This new version of CTSiM also includes two searchable sets of hypertext resources, one for domain information and one for information about use of CT constructs and agent-based modeling. This version also offers students opportunities for checking their science and CT understanding through formative quizzes administered by a mentor agent – Ms. Mendoza. Ms. Mendoza chooses a set of multiple-choice questions based on knowledge needed for the unit the student is working on, and grades students' responses. If a student makes a mistake, she points out the relevant page from the science or CT resources which the student needs to read. Students can choose to take none or multiple formative quizzes in each unit. They can also review their last quiz taken and agent feedback received, or retake the questions they got wrong on their last quiz to improve their scores.

The CTSiM learning curriculum employed in the study comprises four modeling units: (1) Modeling shapes like squares and triangles with equal-length segments, illustrating constant speed, then extending them to increasing and decreasing spirals to model acceleration and deceleration; (2) Modeling roller-coaster behavior as it traverses different segments of a track. An expert simulation is provided to help students build models to match the expected behavior during each segment; (3) Modeling part of a closed fish tank system - a macro-level semi-stable model involving the food chain, respiration, and reproduction processes of fish and duckweed. The non-sustainability of the model (the fish and the duckweed gradually die off) encourages students to reflect on the probable cause and prompts the transition to the next unit involving (4) Modeling bacteria in the waste cycle, which, through stages, help convert the toxic fish waste to nutrients for the duckweed.

3. Method

We conducted a new classroom study using CTSiM with students in a 5th grade class of a middle Tennessee public school (average age of students was 10.5) with 26 students. The study was supervised by the science teacher and one of our researchers. Before the study, students were introduced to the CTSiM system and the modeling units, the CT resources describing how to model given scenarios using an agent-based modeling paradigm and CT constructs. As students worked individually on their modeling tasks, they were periodically reminded to refer to the science and CT resources. In addition, the teacher and the researcher, provided some front-of-the-class and individual help, when students were unsure about how to debug their models.

Students worked in 45-minute daily sessions for 15 days over a span of 3 weeks. On Day 1, students took pre-tests for CT and Kinematics and Ecology science content. They worked on Modeling Units 1 and 2 on days 2-7, before taking the Kinematics post-test and a first CT post-test on day 8. Students then worked on the Ecology units 3 and 4 on days 9-14. They took the Ecology post-test and a second CT post-test on day 15. Each modeling unit included online multiple-choice question pre- and post-quizzes before and after the unit that tested important science and CT facts and concepts linked to that unit. All student actions on the CTSiM system were logged for post-hoc analysis.

3.1 Pre-post assessments for measuring science and CT learning

Pre-post assessments for Kinematics, Ecology and CT tested students' understanding of science concepts and CT skills as well as the ability to solve problems by combining multiple fundamental concepts. The Kinematics pre/post-test assessed whether students understood the concepts of speed, acceleration and distance and their relations. The test required interpreting speed-time graphs and generating diagrammatic representations to explain motion in a constant acceleration field (Basu, et al., 2013; 2014). An example question asked students to diagrammatically represent the time trajectories of a ball dropped from the same height on the earth and the moon, and generate corresponding speed-time graphs. For the Ecology test, questions focused on students' understanding of the role of the species in a fish-tank ecosystem, their interdependence, and how a change in one species affected the others. An example question asked was "*Your fish tank is currently healthy and in a stable state. Now, you decide to remove all traces of nitrobacter bacteria from your fish tank. Would this affect a) Duckweed, b) Goldfish, c) Nitrosomonas bacteria? Explain your answer*."

The CT test required students to generate algorithms for scenarios described in text form using primitives specified in the questions. Simple questions tested use of a single CT construct, while modeling complex scenarios involved use of CT constructs like conditionals and loops and domain-specific constructs. This tested students' abilities to develop meaningful algorithms using programmatic elements like conditionals, loops and variables.

Questions on the online pre-post quizzes tested single science or CT concepts. For example, a quiz question for the fish-micro unit asked: "*Which of the following things does Nitrobacter consume to increase its energy?* Students answered the question picking one of 4-5 choices. These questions were treated as formative assessments, and student responses were graded and feedback was offered when students selected an incorrect answer.

3.2 Assessing students' computational models

We evaluated students' computational models in each unit by comparing against the corresponding expert model. For this comparison, we developed a vector-distance metric (Basu, et al., 2014) to measure the dissimilarity between a student's model and the expert model. This metric is based on a Bag of Words (BoW) (Piech, et al., 2012) representation of the models, in which each procedure (i.e., agent behavior) is analyzed as the set of primitives used to model the procedure. Each primitive was further labeled as primarily computational (e.g., the 'repeat' primitive) or primarily domain-related (e.g. the 'speed' primitive), allowing us to calculate a separate distance measure for computational and domain aspects of the models. The correctness of a procedure was then computed as the size of the intersection of the student and expert models. The sum of procedure correctness scores, normalized by the size of the expert model, provides an overall model correctness measure, bounded between 0 and 1 inclusive.

To account for extraneous primitives, we also calculated an incorrectness measure for each procedure by counting the number of extra primitives in the procedure. Again these procedure incorrectness measures were aggregated and normalized by the size of the expert model to provide the overall model incorrectness score. To determine the overall vector-distance metric, we used a two-dimensional (correctness, incorrectness) vector and calculated its distance to the vector (1,0) that implies a complete BoW match to the expert model. A distance of 0 indicates a perfect BoW match (i.e., the student's model contained all the primitives in the expert model and no extraneous primitives). Using the vector-distance metric, we computed the students' model progression as the intervention progressed. For each change students made to their model, we calculated the distance-to-expert of the resulting intermediate model. An aggregate edit effectiveness measure, i.e., the proportion of model edits made that were effective was computed for each student by unit.

4. Results

We report learning gains, modeling performance metrics and time spent on the science and CT resources to understand the relations between students' actions in the system and their learning gains. On an average, in each unit, students took about 0.8 new quizzes, and retook as quiz 0.43 times.

4.1 Assessing Pre-post Learning Gains

Table 1 presents the results of the paper-based pre-post tests. These results show that the intervention produced significant learning gains in both the science and CT domains. The pre- to post-test gains for the Kinematics and Ecology domains were significant, with effect sizes of 0.34 and 2.65, respectively. Students also gained significant in the CT post-test after Kinematics, and further gained (however, not significantly) after the Ecology unit. These results indicate the synergistic learning of science and CT concepts, and that there was some transfer of CT concepts from one science domain to another.

4.2 Modeling performance and behavior in and across activities

We calculated the vector-distance, effectiveness, and consistency metrics to study the accuracy of students' final models and their model evolution for each activity (see Table 2). When one studies model accuracy over time, the average model accuracy keeps improving from the Roller Coaster unit (average distance from expert model = 0.39) to the Fish macro unit (average distance = 0.3) , and finally to the Fish micro unit (average distance = 0.24). We computed the distance metrics for domain and CT primitives separately and noted the same trend which was not surprising since the use of domain and CT primitives was significantly correlated for all students in each activity. Similarly, the effectiveness and consistency metrics generally improve with time, implying that students made a smaller percentage of incorrect edits in later activities, and when they did, they rectified them more quickly.

Table 1: Paired t-tests showing learning gains for the Kinematics, Ecology, and CT pre-post tests

Domain	Pre-test score (mean, sd)	Post-test score (mean, sd)	*p*-value 2-tailed	Effect Size
Kinematics (max score = 36.5)	13.62 (5.84)	18.38 (7.1)	< 0.05	0.34
Ecology (max score = 32.5)	5.65 (2.85)	19.69 (6.94)	< 0.0001	2.65
Computational Thinking – Post Test 1 (max score = 1 – normalized)	0.34 (0.19)	0.64 (0.14)	< 0.0001	1.80
Computational Thinking – Post Test 2 (max score = 1 – normalized)	0.34 (0.19)	0.69 (0.19)	< 0.0001	1.84

Table 2: Modeling performance and behavior across learning activities

Measures	Roller Coaster unit	Fish-macro unit	Fish-micro unit
Final model distance	.39 (.09)	.30 (.23)	.24 (.37)
Number of model edits	155.0 (63.9)	232.2 (87.7)	134.3 (62.9)
Effectiveness of edits	.38 (.08)	.52 (.07)	.58 (.11)
Consistency of edits	.70 (.15)	.87 (.19)	.86 (.17)

We also studied the relations between students' final models, their pre- to post-test learning gains, and the paths they took to reach their final models. We observed that model accuracy predicts learning gains for the corresponding science content and for CT skills in the later modeling activities after students have become sufficiently familiar with CT constructs (for example, r(micro-final-distance, Ecology gain) = -0.52, $p < 0.01$). Similarly, in latter modeling activities, the effectiveness and consistency of students' edits became strong predictors of final model accuracies for the activities ($p < 0.05$).

4.3 Information acquisition behavior

We also calculated the amount of time students spent on the domain and CT resources in each activity (see Table 3). Students spent a considerable amount of time on the CT resources during their first CTSiM modeling activity, but the use of CT resources dropped to less than 10% of the initial reading time for the remaining units. This does not seem surprising since students initially needed to learn how to use the CT constructs and the agent-based framework for building computational domain models, but after gaining an initial understanding, these abilities seem to transfer easily to other domain models.

Table 3: Time (in seconds) spent in domain and CT resources for different activities

Resources	Units				
	Constant Shape Drawing	Variable Shape Drawing	Roller Coaster	Fish-macro	Fish-micro
Domain	742.9 (262.2)	508.0 (194.8)	427.6 (251.4)	1160.1 (550.7)	1045.9 (509.1)
CT	1221.6 (1359.5)	92.08 (110.3)	44.3 (86.8)	45.7 (69.8)	34.3 (82.6)

In case of domain resources, the trend differs. For the three Kinematics units (constant speed and variable speed shape drawing), the time spent on domain resources dropped, but at a much smaller rate than the time spent on CT resources. When students switch to modeling ecological processes (the fish macro and the fish micro units), there is a large increase in time spent on the resources (the time more than doubles). On further analysis, the difference in resource access time makes sense. The resources for the shape drawing units describe the relation between acceleration, speed and distance, and the meaning of turn angles. The roller coaster computational model is more complex, and combines speed up, slow down, and move at a constant speed functions. In each unit, new concepts are included that build on concepts presented in the previous unit. Therefore, the read time decreases, but not by large amounts. When the switch is made from the kinematics to the ecology units, there is a considerable increase in the number of new concepts required to build successful models. In addition, the classroom teacher provided less instruction on the specifics of the ecological models, and asked students to read the resources carefully when constructing their models.

We also studied whether students' time spent on reading resources in an activity was related to the accuracy of their models built in that activity. We find that students' reading times correlate well with their final model distance scores (e.g., r(micro-distance, domain-reading-time) = 0.41, $p < 0.05$).

5. Discussion and Conclusion

In this paper, we present a new version of the CTSiM system – a learning environment for fostering synergistic learning of middle school science and CT. To help students with their individual learning, we have added science and CT resources, and introduced formative assessments to allow students to check their understanding of science and CT concepts during model building and refinement. Our in-class study with 5th grade students with the new version of CTSiM produced significant science and CT learning gains with large effect sizes, and showed that computational concepts, once learned, could form the basis for modeling in other science domains. The gains were further supported by additional metrics that showed students developed the abilities to build more accurate models and make more consistent model edits. Their use of resources was also consistent with the complexity of the units they worked on. Further, model accuracy is predictive of learning gains, edit effectiveness and consistency.

Our current findings indicate that *formative assessments* and corresponding adaptive scaffolding can play an important role in supporting student learning in OELEs (Azevedo & Hadwin, 2005; Land, 2000). This study did not produce conclusive evidence of the effectiveness of formative assessments and adaptive scaffolding in making students better modelers and science learners, but, in the next iteration of CTSiM, our plans are to track students' model building activities to dynamically scaffold students in their concept learning, model building, and model debugging tasks.

Acknowledgements

This work was supported by the NSF (NSF Cyber-learning grants # 1124175 and #1441542).

References

Azevedo, R., & Hadwin, A. F. (2005). Scaffolding self-regulated learning and metacognition–Implications for the design of computer-based scaffolds. *Instructional Science*, 33(5), 367-379.

Basu, S., Dukeman, A., Kinnebrew, J., Biswas, G., & Sengupta, P. (2014). Investigating student generated computational models of science. In *Proceedings of the 11th International Conference of the Learning Sciences* (pp. 1097-1101). Boulder, CO, USA.

Basu, S., Dickes, A., Kinnebrew, J. S., Sengupta, P., & Biswas, G. (2013). CTSiM: A Computational Thinking Environment for Learning Science through Simulation and Modeling. In *Proceedings of the 5th International Conference on Computer Supported Education* (pp. 369-378). Aachen, Germany.

Grover, S., & Pea, R. (2013). Computational Thinking in K–12. A Review of the State of the Field. *Educational Researcher*, 42(1), 38-43.

Guzdial M. (1995) Software-realized scaffolding to facilitate programming for science learning. *Interactive*

Land, S. M. (2000). Cognitive requirements for learning with open-ended learning environments. *Educational Technology Research and Development*, 48(3), 61-78.

National Research Council. (2011). A framework for K-12 Science Education: Practices, Crosscutting Concepts, and Core Ideas. Washington, DC: The National Academies Press.

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.

Wilensky, U., & Reisman, K. (2006). Thinking like a wolf, a sheep or a firefly: Learning biology through constructing and testing computational theories - An embodied modelling approach. *Cognition & Instruction*, 24(2), 171-209.

Wing, J. (2011). Research notebook: Computational thinking—What and why? *The Link Magazine*, Spring Carnegie Mellon University, Pittsburgh, USA. Retrieved from http://link.cs.cmu.edu/article.php?a=600