Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16)

Using Multiple Representations to Simultaneously Learn Computational Thinking and Middle School Science

Satabdi Basu, Gautam Biswas, and John S. Kinnebrew
Institute for Software Integrated Systems
Vanderbilt University
Nashville, TN, USA.
{Satabdi.basu@vanderbilt.edu, gautam.biswas@vanderbilt.edu, john.kinnebrew@gmail.com}

Abstract

Computational Thinking (CT) is considered a core competency in problem formulation and problem solving. We have developed the Computational Thinking using Simulation and Modeling (CTSiM) learning environment to help middle school students learn science and CT concepts simultaneously. In this paper, we present an approach that leverages multiple linked representations to help students learn by constructing and analyzing computational models of science topics. Results from a recent study show that students successfully use the linked representations to become better modelers and learners.

Introduction

Research in science education has emphasized the importance of engaging learners in the practices of developing and using models to help them understand how scientific knowledge is constructed, evaluated, and communicated (Duschl, Schweingruber, & Shouse, 2007). Computational Thinking (CT) is being recognized as a vital ingredient to support such learning (Barr & Stephenson, 2011; Grover & Pea, 2013). Several of the epistemic and representational practices central to the development of expertise in science disciplines (e.g., defining problems, developing and using models, and designing and verifying solutions) are primary components of CT (Wing, 2006). In order to support middle school students' synergistic learning of science and CT concepts, we have developed the Computational Thinking using Simulation and Modeling (CTSiM) learning environment (Basu et. al., 2014; Sengupta et. al, 2013).

CTSiM provides an agent-based, visual programming platform where students build simulation models of science phenomena using an agent-based framework (Wilensky, 1999), and analyze and compare their model behaviors against matched expert simulations to refine their models. In this paper, we present our latest version of CTSiM, where we have extended the computational/simulation modeling representation to include a linked conceptual modeling representation to better support students' modeling and learning tasks. Conceptual modeling helps students identify the primary agents, their properties and behaviors, and relevant environment elements in the domain of study. Further, students conceptualize agent behaviors as sense-and-act processes. In concert, students build their computational models to describe agent behaviors using a visual block-structured language. The set of programming constructs or blocks include domain-general CT (e.g., assignments, conditionals, loops) and domain-specific (e.g., 'energy' in ecological models, 'speed up' in kinematics models) constructs (Sengupta, et al., 2013). In previous work, we have demonstrated that the domain-specific computational modeling language helps leverage students' intuitions and science knowledge while synergistically supporting learning science and CT concepts across different domains (Basu et. al., 2014).

In this paper, we formally describe the linked modeling representations as different levels of abstraction in CTSiM modeling activities. Using results from a 6th-grade classroom study, we show how students combine the use of these representations while building, analyzing, and verifying their models to learn science and CT concepts. Analysis of students' modeling behavior shows that when appropriately scaffolded, their use of the linked representations improves over time. Also, they develop the ability to decompose complex models into parts and use a judicious combination of the conceptual and computational representations to construct correct domain models. In addition, their modeling

Copyright © 2016, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

behavior using multiple linked representations is strongly correlated with their science learning.

CTSiM Approach: Learning by Modeling

CTSiM adopts a learning-by-modeling pedagogical approach, where students alternate between phases of model building, simulation, verification, and refinement tasks. They also have access to searchable hypertext resources containing information about relevant science and CT concepts. As students build their conceptual and computational models, they can visualize their model behaviors as NetLogo simulations (Wilensky, 1999), and verify their evolving models by comparing the model behaviors against a matched 'expert' simulation. They do not have access to the expert computational model, but they can study and analyze the differences between the simulation results to guide them in improving their models. Figure 1 depicts the CTSiM learning-by-modeling process.

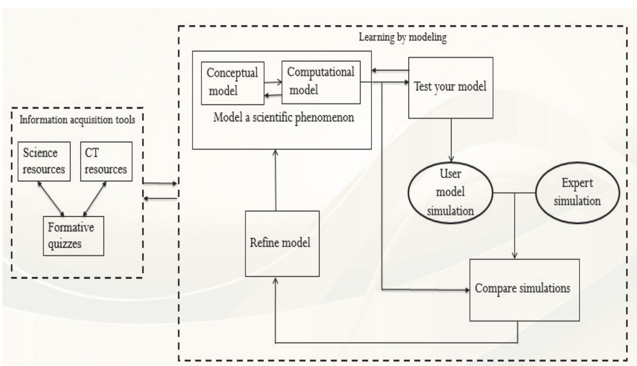

Fig 1. *Learning using CTSiM*

Currently, CTSiM includes three primary modeling activities in Kinematics and Ecology. Activity 1 models a roller coaster (RC) car moving along a track with four segments - *up at constant speed* (pulled by a motor); *down* (free fall); *flat* (constant speed); and *up against gravity*. This activity targets the Kinematics concepts of speed, distance, and acceleration and their relations. In Activities 2 and 3, students progress from modeling single agent behaviors in the RC activity to modeling multiple agents with multiple behaviors as they model the ecological processes in a fish tank system. In Activity 2, students build a *macro*-level, semi-stable model of a fish tank with two types of agents: fish and duckweed, and behaviors associated with the food chain, respiration, locomotion, and reproduction of these agents. Since the waste cycle is not modeled, the build-up of toxic fish waste results in the non-sustainability of the macro-model (the fish and the duckweed gradually die off). In Activity 3, students address the non-sustainability by introducing micro-level entities, i.e., Nitrosomonas and Nitrobacter bacteria, which together support the waste cycle, i.e., converting the ammonia in the fish waste to nutrients (nitrates) for the duckweed. The plots generated by the simulation models help students gain an aggregate level understanding of the different cycles and their role in establishing the interdependence and balance among the different agents in the system.

Multiple linked representations to support building the simulation model

CTSiM supports students' model building activities with two linked representations. Multiple external representations (MERs) are known to help in developing a deeper understanding of domain concepts that would be difficult to achieve with a single representation (Ainsworth, 2006). The ability to construct and switch between multiple perspectives in a domain helps learners build abstractions that are fundamental to successful learning in the domain (Ainsworth & van Labeke, 2004). Furthermore, insights achieved through the use of MERs increases the likelihood of transfer to new situations (Bransford & Schwartz, 1999).

However, studies on the benefits of MERs have produced mixed results for novice learners, possibly due to the cognitive load imposed by dealing with MERs (Mayer & Moreno, 2002; Ainsworth, 2006). Learners have to understand the constructs and semantics associated with each representation, while also discovering the relations between these representations. Studies have shown that learners tend to treat representations in isolation and find it difficult to relate, translate between, and integrate information from MERs, (van der Meij & de Jong, 2006). To derive benefits from MERs, learners need to be supported. Some common forms of support include integrated presentation of the MERs with dynamic linking or translation between them (Ainsworth, 2006; Goldman, 2003).

CTSiM operationalizes the important CT concepts of abstraction and decomposition with two complementary linked modeling representations that support modeling at different levels of abstraction. Students start constructing an abstract conceptual model of the domain, which they can leverage to build the computational models of individual behaviors. Though this implies a hierarchical structure between the two representations, students have the freedom to switch between the representations as they construct and refine their models in parts. CTSiM provides support for integrating and maintaining correspondence between the representations by (1) the modeling interface, which provides an integrated presentation that highlights the links between the representations, and (2) individualized feedback from a mentor agent, when students are unable to make progress in their modeling tasks.

In the conceptual model representation, students use a visual editor to identify the primary agents and environmental elements in the domain of study, along with the relevant properties associated with these entities. Students also identify agent behaviors and represent the behaviors in terms of *sensed* and *acted-upon* properties. For example, in the fish-macro activity, 'fish' represents an agent with properties like 'hunger' and 'energy' and behaviors like 'feed' and 'swim', while 'water' is an environment element with properties like 'dissolved oxygen' and 'cleanliness'. The 'fish-feed' behavior senses the properties 'fish-hunger' and 'duckweed-existence', and acts on properties like 'fish-energy'. However, this representation abstracts several details like how and when the different properties are acted on in the various agent behaviors.

Instead, these details are modeled in the computational model representation. For this purpose, students use a linked visual interface that supports the selecting and dragging of primitives (blocks) from a palette with domain-specific primitives (e.g., "speed-up" in kinematics, "wander" in ecology) and domain-general CT primitives (e.g., conditionals and loops). This domain-specific modeling language with domain-general computational constructs helps emphasize the domain-specific science concepts and the generality of CT concepts across science domains, helping students synergistically learn science and CT concepts.

The properties specified in the sense-act conceptual model representation for an agent behavior determine the set

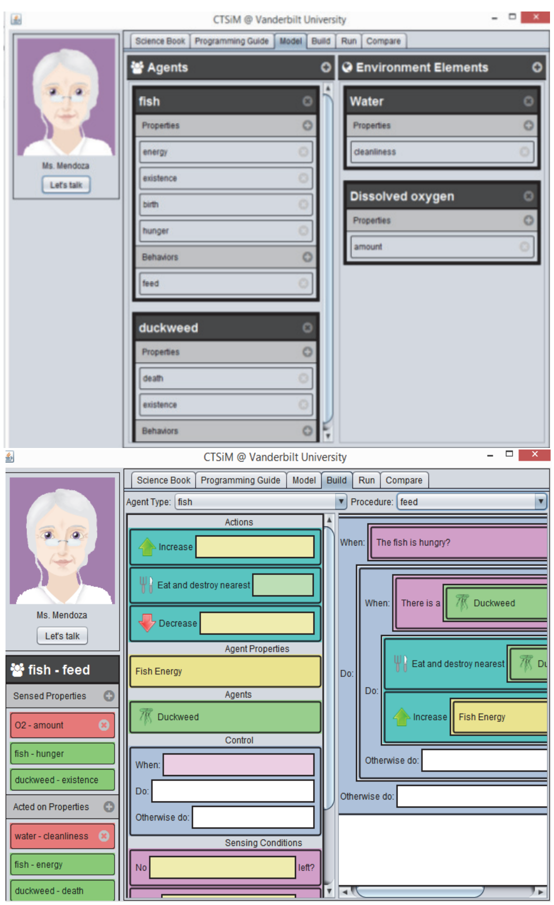

Fig 2. *Conceptual and computational modeling representations*

of domain-specific primitives available in the palette for modeling the behavior. This dynamic linking helps students gain a deeper understanding of the representations and their relationships. For example, the 'wander' block is available in the 'fish-swim' behavior only if 'fish-location' is specified as an acted on property for the behavior. CTSiM adopts a single internal representation for specifying the agent-based conceptual and computational modeling constructs, and a sense-act framework that help students focus on concepts associated with a specific science topic, while also accommodating CT constructs that apply across multiple science domains.

Figure 2 depicts the modeling representations. The top screenshot is a part of the conceptual modeling interface. The second screenshot represents a combined conceptual-computational interface for modeling agent behaviors ('fish-feed' in this case). The leftmost panel depicts the sense-act conceptual representation, while the middle panel shows the computational palette, and the right panel contains the student-generated computational model. The side-by-side placement of the representations provides integrated presentation support and is based on the fact that learners find it easy to understand physically-integrated material rather than separately presented material (Chandler & Sweller, 1992).

To further aid the integration, the red/green coloring of the sense-act properties provides students with visual feedback about how closely their computational models for an agent behavior correspond to their conceptual model for that behavior. Initially, the sense-act properties are colored red. As students build their computational model and add sensing and action blocks corresponding to these properties, they change color from red to green (another example of support provided by the dynamic-linking). For example, in Figure 2, the student has conceptualized that O_2-amount needs to be sensed for the fish-feed behavior. However, the computational model does not include this information and hence the property is colored red. In such cases, students can verify individual agent behaviors and decide how to refine their computational and/or conceptual models.

Scaffolds for conceptual and computational modeling and their integration

Besides the inherent support provided by the interface design that emphasizes the links between the conceptual and computational representation as presented above, we also designed a set of adaptive scaffolds to support students when they face persistent problems in their modeling tasks. The scaffolds, delivered through a text-based conversational dialogue initiated by the mentor agent in the system, point to the problems observed in students' models (this is done by tracking the students' actions and their evolving models) and suggest appropriate strategies to help them overcome their difficulties. The strategies implemented in the current version of CTSiM include: (1) pointers to the science resources to help students acquire relevant information needed to build or refine a part of a model; and (2) suggestions for simulating the model in parts to make it easier to identify discrepancies in agent behaviors.

The designed scaffolds are delivered to the students following a top-down approach, first pointing out problems with the conceptual model before focusing on issues of coherence between the conceptual and computational representations, and finally identifying problems specific to the computational model like incorrect arrangement of blocks. Examples of scaffolding dialogues are (1) "*The fish-breathe behavior requires interaction of the fish with other entities. Have you considered all the entities in this science topic*?"; (2) "*You have all the necessary blocks for the rollercoaster-update speed behavior, but are you sure that all the actions occur under the right set of conditions*?"; and (3) "*You have unused properties colored red in the fish-feed behavior. Do you want to use them in your program or do you want to delete the properties*?"

Method

We report a recent CTSiM study with 52 students from two 6th grade sections (average age=11.5). This study was part of a larger controlled study with 98 students grouped into two conditions – an experimental group that received the adaptive scaffolding and a control group that did not have access to the adaptive scaffolding. In this paper, we limit our discussions to the students in the experimental group who received the adaptive scaffolding, and their use of the linked representations for model-building in CTSiM.

Each student worked individually on the three modeling activities described in the previous section. The study was run daily over a span of three weeks during class science periods. During this period, the science teachers for both the sections ensured that they did not provide any instruction on or assign any activities related to the Kinematics and Ecology concepts covered in the CTSiM activities. On Day 1, students took paper-based tests in Kinematics, Ecology, and CT. The Kinematics questions tested students' understanding of the relationships between distance, speed, and acceleration. The Ecology questions focused on the interdependence among the species in an ecosystem, along with the energy, respiration, and the waste cycles. CT skills were assessed by asking students to predict program segment outputs, and model scenarios using CT constructs.

On day 2, students were introduced to agent based modeling and the CTSiM system. They practiced a simple shape drawing activity that modeled the relations between distance, speed, and acceleration. Students then worked on the Rollercoaster unit through Day 6. They took paper-based post-tests on Kinematics and CT on Day 7. On days 8-12 students worked on the Ecology modeling activities. They took their Ecology post-tests and CT-final post-tests on Day 13. On Day 14, students worked on a paper-based wolf-sheep-grass ecosystem modeling activity as a transfer task.

In this paper, we assess students' pre-post learning gains, modeling performance and modeling behaviors (from data collected in log files) to answer the following research questions:

1. *Did the intervention help students learn the desired science and CT concepts?*
2. *How did students use and integrate their conceptual and computational modeling activities in developing their simulation models?*
3. *How did students' modeling performances and behaviors relate to their science learning?*

We assess students' conceptual and computational modeling performances for an activity in terms of the distances between the student models and the corresponding expert models. A model distance of 0 implies that the student model is a perfect match to the expert model (no extra and incorrect constructs). A more comprehensive definition of the distance metric can be found in Basu et. al. (2014).

Students' modeling behaviors during an activity are characterized by the rate of progress in building their conceptual and computational models, and how they combine the two representations to build their models. We describe model evolution using 3 metrics: (1) *Effectiveness*- the proportion of model edits that bring the model closer to the expert model; (2) *Slope* – the rate and direction of change in the model distance as students build their models; and (3) *Consistency* – How closely the model distance evolution matches a linear trend. We assess integration of the conceptual and computational representations with 3 other metrics: (1) We look at activity chunks of each type and use the *total number of chunks* as a measure of how many times a student switched between the two representations; (2) The *average sizes* of the conceptual and computational modeling *chunks*, and their *ratio* constitutes the second metric; and (3) the fraction of conceptual edits that were followed by *related* (*coherent*) computational edits defines the third metric.

Results

Overall, the pre- and post-test results showed that the intervention produced strong learning gains for the science and CT concepts (research question 1). Table 1 shows the results of a one-way repeated measures ANOVA of pre-post differences. The gains were significant at the $p<0.0001$ level with large effect sizes (Cohen's *d*).The CT post-scores in Table 1 refer to the scores on the final CT test administered after the Ecology unit. The intermediate (post-kinematics) CT test results also showed significant gains ($p<0.0001$, effect size=0.82).

Modeling performance and behavior

For the analysis supporting our second research question, we first studied students' conceptual and computational modeling behaviors and performance separately (Table 2), and then computed the integration metrics (Table 3). Table 2 shows that both the conceptual and computational modeling behavior metrics, i.e., effectiveness, slope and consistency improved significantly from the RC activity to the fish-micro activity with significance levels of $p<0.001$.

Table 1: Pre-post gains [mean (s.d.)] for science and CT content

	Pre	Post	*p*-value	Effect size
Kinematics (max=45)	16.65 (6.61)	22.38 (6.39)	<0.0001	0.88
Ecology (max=39.5)	9.39 (4.47)	27.91 (6.70)	<0.0001	3.25
CT (max=60)	22.72 (7.68)	32.24 (5.86)	<0.0001	1.39

Table 2: Modeling performance and behavior for conceptual and computational modeling individually [mean (s.d.)]

		RC model	Macro model	Micro model
Conceptual modeling	Final distance	0.2 (0.2)	0.13 (0.13)	0.14 (0.08)
	Effective-ness	0.57 (0.04)	0.59 (0.04)	0.68 (0.06)
	Slope	-0.003 (0.003)	-0.002 (0.002)	-0.005 (0.003)
	Consistency	0.3 (0.22)	0.59 (0.31)	0.8 (0.22)
Computa-tional modeling	Final distance	0.24 (0.25)	0.09 (0.1)	0.04 (0.08)
	Effective-ness	0.43 (.08)	0.58 (.08)	0.69 (.11)
	Slope	–0.006 (.005)	–0.004 (.002)	–0.009 (.004)
	Consistency	0.6 (.25)	0.95 (.04)	0.95 (.05)

Table 3 shows how students combine the conceptual and computational representations in each activity, and how these metrics change across the activities. Initially, the average size (number of edits) of each editing chunk was large and the number of switches between the conceptual and computational model editing was small. In later activities, the chunk sizes decreased and the number of switches increased, which implies that students decomposed their modeling tasks into smaller units. We computed the normalized ratio of conceptual and computational chunk sizes (normalized it by the ratio of the size of the expert models) to define another measure of integration among the modeling representations. The normalized ratio for the RC model is 1.1 implying it is similar to the ratio for the conceptual and computational components of the expert RC model. This ratio increased for the two later activities (increasing to 2 for the fish macro model) implying that students' conceptual edits increased as compared to their computational edits with respect to the corresponding expert models. Perhaps, the complexity of the ecology domain, as well as students' lower prior knowledge in the domain (lower pre-test scores in Table 1) resulted in the students spending more effort (i.e., more edits because they made more errors) in conceptualizing the models (multiple entities, their properties, and behaviors) than in the kinematics unit. Also, as the complexity of the model increased, the average size of both the conceptual and computational model building chunks decreased, implying that the increased complexity led students to decompose their model building into smaller chunks, and switch more often between the two linked representations in constructing the model chunks. For all activities, the median conceptual edit occurred at around 40% of all the model edits, meaning more of the initial model edits were conceptual rather than computational. This provides indirect evidence that the students employed a top-down modeling approach, starting with the conceptual model and then switching to constructing the corresponding segments of the computational model.

Table 3: Modeling behavior for integrating representations

	RC activity	Macro activity	Micro activity
Number of chunks	33.23 (11.57)	93.52 (30.11)	56.17 (13.56)
Average size of conceptual chunks	8.24 (2.44)	8.12 (3.33)	5.65 (1.6)
Average size of computational chunks	7.92 (2.78)	5.11 (1.25)	4.2 (1.26)
Normalized ratio of conceptual to computational chunk sizes	1.1 (.52)	2.02 (.87)	1.38 (.42)
Median conceptual edit with respect to all edits	0.37 (.09)	0.43 (0.05)	.41 (.05)
Conceptual edits followed by coherent computational edits	.3 (.08)	.33 (.11)	.56 (.12)

To better understand how students coordinated their conceptual and computational modeling activities, we also analyzed the relationship between edits in the two representations. Conceptual and subsequent computational edits are 'coherent' if both are related to the same aspect of the model being constructed. For example, if a student added 'breathe' as a fish agent behavior in their conceptual model, then any subsequent edits to the blocks in the fish-breathe computational model were defined to be 'coherent' with the conceptual edit. Similarly, if a student added 'fish-hunger' as a sensed property in their conceptual model for a particular agent behavior, adding a sensing block like "Is the fish hungry?" to the computational model for the behavior is considered 'coherent'. Only additions to the conceptual model were considered in computing coherence relations since coherence is characterized by computational edits that are linked to previous conceptual edits. The results showed significant improvement in this metric from the RC to the fish-micro activity (p<0.0001), and that about 50% of these coherent computational edits were made immediately following the conceptual edits in all the activities.

Effects of modeling performance and behavior on learning

To investigate the third research question, we analyzed the correlations between the modeling measures for each activity (described in Tables 2 and 3) and students' post-test scores for the corresponding science domain. We did not find any significant correlations between students' modeling measures in the RC activity and their Kinematics posttest performance. A likely reason is that the RC conceptual representation, with a single agent type, did not provide a lot of scaffolding in helping design the corresponding computational models. Therefore, the benefits of the linked representation are not as apparent. Besides the students may not have become proficient with the representations in activity 1, therefore, the linked representations did not help the students to better understand domain knowledge. However, Table 4 shows that the modeling metrics in the fish-macro and fish-micro activities were significantly correlated with their Ecology post-test scores.

Table 4: Correlations of modeling performances and behaviors with ecology learning (*p<0.05; **p<0.01)

	Macro unit	Micro unit
Final conceptual distance	-0.476**	-0.393**
Conceptual effectiveness	0.334*	0.275*
Conceptual slope	-0.204	-0.286*
Conceptual consistency	0.3116*	0.281*
Final computational distance	-0.393**	-0.182
Computational effectiveness	0.225	0.131
Computational slope	-0.265	-0.155
Computational consistency	0.264	0.104
Conceptual chunk size	-0.42**	-0.29*
Computational chunk size	-0.21	-0.3*
Coherent edits	0.293*	0.275*

The correlations between macro and micro final conceptual model distances, and the macro final computational model distance with ecology post-test scores were significant (lower distance indicates better models, which was associated with higher post test score). Also, nearly all of the conceptual model evolution metrics for the fish-macro and fish-micro activities were correlated with ecology post-test scores (higher conceptual effectiveness and consistency and more negative slopes were associated with higher post test scores).

In terms of linked representation integration metrics, the fraction of coherent edits in both the macro and micro units were significantly correlated to ecology post test scores (higher fraction of conceptual edits with coherent computational edits was associated with higher post test scores). Though the ratio of chunk sizes was not significantly correlated with learning, we found the macro average conceptual chunk size to be significantly correlated with post test scores ($r = -0.42$, p<0.01), and the micro conceptual and computational chunk sizes were also significantly correlated with post test scores ($r= -0.29$, p<0.05; and $r = -0.3$, p<0.05), meaning smaller chunk sizes implied higher post test scores. With respect to integrating the two representations, these results suggest that effective coordination between the linked representations appears to have an important positive effect on science learning. Specifically, decomposing the modeling task and going back-and-forth between representations in relatively small sized chunks that were coherent (as measured by the coherence edit proportion) appear to be useful behaviors that supported greater learning. These are especially promising results given the general positive trends in these measures over the course of the CTSiM progression of modeling activities. In summary, we find that the accuracy of the conceptual and computational models students build, the evolution of the models, as well as how well students integrate their activities with respect to the two modeling representations, are all strong indicators of learning science content while working in the CTSiM environment.

Discussion and Conclusions

We have presented a learning-by-modeling approach that combines two linked modeling representations and ideas from CT (model building by decomposition, problem solving, monitoring evolving models, and verifying model behaviors) for teaching middle school students about science topics and CT concepts synergistically. With appropriate support provided by the modeling interfaces that highlight the relations between the two representations combined with adaptive scaffolding when students face difficulties, we have demonstrated the effectiveness of using multiple representations in synergistic learning of science and CT concepts. In fact, we found that the adaptive scaffolding was critical for enabling effective integration and use of the two modeling representations. Students in the control group that did not receive adaptive scaffolding (see Method section) also showed science and CT learning gains, but their gains were significantly lower than that of students in the experimental group whose results have been discussed in this paper. Also, compared to the control group, the experimental group showed significantly better modeling performance with each of the representations, and better modeling behaviors with respect to integrating the representations and maintaining correspondence between them. Our results thus agree with the existing literature on MERs (Ainsworth, 2006; Mayer & Moreno, 2002) that hypothesize that MERs are powerful tools, but the representations need to be carefully integrated and learners need to be supported to help them manage their cognitive load so that they may derive the increased understanding from these representations. In our case, the scaffolds afforded by the system linking the conceptual and computational representations are further aided by the adaptive scaffolding that guides students in effective use of the two closely linked representations.

In addition, we have shown that the two linked representations promote effective modeling behaviors (such as a 'divide and conquer' strategy) on complex modeling tasks, even though this strategy was not explicitly taught or required for the model building tasks. Our results also show that students' abilities to integrate the two representations effectively and progressively build more accurate models improved as they worked on more activities. Equally important, our results showed that proper use of the linked representations also led to better learning of science domain content as demonstrated by the pre- to post-test learning gains. Overall, the linked representations coupled with adaptive scaffolding not only helped improve students' learning of science concepts, but they also helped develop important CT concepts related to model building, model analysis and verification, and problem solving.

As future work, we plan to study in greater detail how students combined their model building tasks with other tasks, such as information acquisition and model verification, and how these may have been influenced by the linked representations. We also plan to analyze how the individual scaffolds helped students learn from the coupled modeling representations. Also, how often were the different scaffolds provided, and how students responded to them. These analyses should help us understand how to scaffold students more effectively in such scenarios.

Last, we would like to conduct more definitive studies using a control-experimental design to gain a better understanding of the role of multiple representations in learning by modeling and problem solving. For such experiments, the control condition would work with a single modeling representation, and the differences in learning behaviors and performance would provide clear evidence for and against using multiple representations.

Acknowledgments

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

References

Ainsworth, S. (2006). DeFT: A conceptual framework for considering learning with multiple representations. *Learning and Instruction*, *16*(3), 183-198.

Ainsworth, S. E., & Van Labeke, N. (2004). Multiple forms of dynamic representation. Learning and Instruction, 14(3), 241-255.

Barr, V., & Stephenson, C. (2011). Bringing computational thinking to K-12: what is Involved and what is the role of the computer science education community? *ACM Inroads*, *2*(1), 48-54.

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.

Bransford, J. D., & Schwartz, D. L. (1999). Rethinking transfer: A simple proposal with multiple implications. Review of Research in Education, 24, 61-100.

Chandler, P., & Sweller, J. (1992). The split-attention effect as a factor in the design of instruction. British Journal of Educational Psychology, 62(2), 233e246.

Duschl, R. A., Schweingruber, H. A. & Shouse, A. W. (2007). *Taking Science to School: Learning and Teaching Science in Grades K-8*. National Academies Press.

Goldman, S. R. (2003). Learning in complex domains: When and why do multiple representations help?. *Learning and instruction*, *13*(2), 239-244.

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

Mayer, R. E., & Moreno, R. (2002). Aids to computer-based multimedia learning. Learning and Instruction, 12(1), 107e119.

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.

van der Meij, J., & de Jong, T. (2006). Supporting students' learning with multiple representations in a dynamic simulation-based learning environment.*Learning and Instruction*, *16*(3), 199-212.

Wilensky, U. (1999). *NetLogo.* Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL. (http://ccl.northwestern.edu/netlogo).

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, *49*(3), 33-35.