Providing Adaptive Scaffolds and Measuring Their Effectiveness in Open Ended Learning Environments

Satabdi Basu, Vanderbilt University, satabdi.basu@vanderbilt.edu
Gautam Biswas, Vanderbilt University, gautam.biswas@vanderbilt.edu

Abstract: Open ended learning environments (OELEs) offer students powerful learning opportunities, but managing the available tools and choices is challenging for novice learners. Adaptive scaffolding can support learners, but it requires understanding and responding to learner actions and strategies. We discuss a generalized adaptive scaffolding framework for OELEs based on a task and strategy model, and effectiveness and coherence measures for evaluating learner proficiencies for tasks and strategies. We apply this framework to support learners in CTSiM (Computational Thinking using Simulation and Modeling), a learning-by-modeling OELE for synergistic learning of science and computational thinking (CT). The effectiveness of our approach is demonstrated by a classroom study with two conditions. Students in the adaptive scaffolding condition showed a better understanding of science and CT concepts, built more accurate models, used better modeling strategies, and transferred modeling skills better to new scenarios than students in the control condition who received no adaptive scaffolding.

Keywords: open ended learning environments, learning by modeling, computational thinking, modeling and simulation, science education, learner modeling, adaptive scaffolding

Introduction

Open-ended learning environments or OELEs (Land, 2000) are learner centered computer environments designed to support thinking-intensive interactions with limited external direction. They typically provide a learning context and a set of tools to help students explore, hypothesize, and build solutions to authentic problems. The complex nature of the problems requires students to develop strategies for decomposing their problem solving tasks, developing and managing the accompanying plans, and monitoring and evaluating their evolving solutions. Thus, OELEs offer powerful learning opportunities for developing metacognitive and self-regulation strategies (Bransford &Schwartz, 1999). However, learning in OELEs is challenging for novices who may lack proficiency in using the system's tools, resulting in adoption of suboptimal learning strategies. Adaptive scaffolding may help learners overcome these difficulties (Puntambekar and Hubscher, 2005).

Many OELEs provide non-adaptive supporting tools like guiding questions, argumentation interfaces, workspaces for structuring tasks, and data comparison tools. As Puntambekar and Hubscher (2005) point out, such scaffolding tools support student learning, but they neglect important features of adaptive scaffolding such as ongoing diagnosis, calibrated support, and fading. Even in OELEs with adaptive scaffolding, few of them provide scaffolds that target students' understanding of domain knowledge, cognitive processes, and metacognitive strategies in a unified framework. MetaTutor (Azevedo, 2005) measures student behaviors using factors, such as the number of hypermedia pages visited and the length of time spent on each page, to decide when to provide adaptive scaffolds, e.g., "*You should re-read the page about the components of the heart*". In Ecolab (Luckin and du Boulay, 1999), the scaffolding agent intervenes when students specify an incorrect relationship in their models and provides a progression of hints, each more specific than the previous one, with the final hint providing the answer. In Co-Lab (Duque et. al., 2012), the system provides feedback on students' models and work processes, but is limited to reminding students about model building and testing actions not taken.

In this work, we have developed a task- and strategy-based modeling framework combined with coherence analysis to interpret and analyze students' actions (Segedy et. al. 2015) in Computational Thinking using Simulation and Modeling (CTSiM) – a learning-by-modeling OELE that we have developed to support synergistic learning of science and computational thinking (CT) in middle school science classrooms (Basu et. al., 2014; Sengupta et. al., 2013). We use the framework to determine need for scaffolding in CTSiM, and report results from a classroom study where a group of students used CTSiM with adaptive scaffolding and another group used it without the adaptive scaffolding. The effectiveness of our scaffolding approach is demonstrated in terms of students' science and CT learning, online modeling performance, learning strategies employed, and transfer of modeling skills outside the CTSiM OELE.

The CTSiM learning environment

The CTSiM environment (Basu et. al., 2014; Sengupta et. al., 2013) adopts an agent-based, learning-by-modeling approach where students' model building activities are supported by two linked representations for conceptual and computational modeling. In the abstract conceptual model representation, students use a visual editor to identify the primary agents and environmental elements in the domain of study, along with their relevant properties. Students also identify agent behaviors and represent them using a sense-act framework by specifying which properties need to be sensed in order for the behavior to occur, and which properties will be acted upon in the behavior. For example, in one of the activities where students model a fish tank, '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 'cleanliness' and "dissolved oxygen." The 'fish-feed' behavior senses the properties 'fish-hunger' and 'duckweed-existence', and acts on properties like 'fish-energy'. However, this representation abstracts details like how and when the different properties are acted on. These details are captured in the computational models, where students use a visual programming environment, and add and arrange provided blocks from a palette to create their models. The programming blocks can be domain-specific (e.g., "speed-up" in kinematics, "feed" in biology) or domain-general (e.g., conditionals and loops). The properties specified in the sense-act conceptual model for a behavior determine the set of domain-specific blocks available in the palette for the behavior. This dynamic linking helps students gain a deeper understanding of the representations and their relations. For example, the 'wander' block is available in the palette of available blocks for the 'fish-swim' behavior only if 'fish-location' is specified as an acted on property for the behavior.

Figure 1 represents the "Build" interface for modeling agent behaviors ('fish-feed' in this case). The leftmost panel depicts the sense-act conceptual representation, 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 is deliberate to emphasize their connectedness. To further aid the integration, the red/green coloring of the sense-act properties provides visual feedback about the correspondence between students' conceptual and computational models for an agent behavior. Initially, all the properties are colored red. As students add 'sense and act' blocks corresponding to the properties, the properties change color from red to green. For example, Figure 1 specifies O_2-amount as a sensed property for the fish-feed behavior. However, the computational model does not include O2-amount 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.

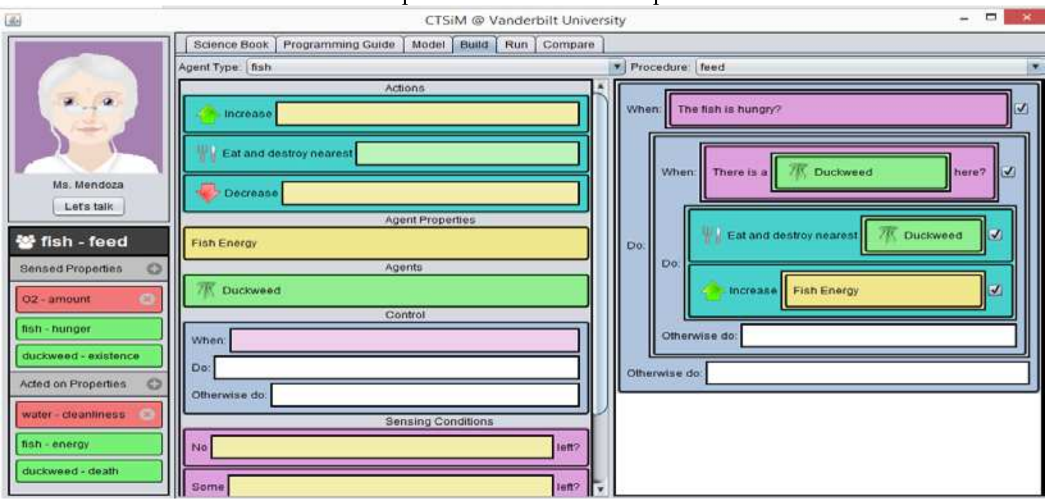

Figure 1. The linked conceptual-computational "Build" interface for modeling agent behaviors.

As students construct their models, they can visualize their model behaviors as NetLogo simulations (Wilensky, 1999), and verify their evolving models (the entire model or a subset of agent behaviors) by comparing the model behaviors against a matched 'expert' simulation. They do not have access to the expert computational model, but can analyze the differences between the simulation results to guide them in improving their models.

CTSiM also provides two sets of searchable hypertext resources, one with information about the science topic being modeled, and the other with information about agent-based conceptual and computational modeling. Students can also check their understanding of science and CT concepts by taking formative quizzes administered by a mentor agent in the system named Ms. Mendoza. The mentor grades students' responses to the multiple-choice type quiz questions and suggests resource pages to read in case of incorrect student responses.

A generalized scaffolding framework for OELEs

We develop a theoretical framing for our generalized scaffolding framework for OELEs and describe how we apply it to the CTSiM environment. Some of the distinguishing features of our framework are as follows:

1. Tracking and interpreting learner behaviors using a task and strategy model
2. Determining the effectiveness of tasks and strategies using coherence metrics based on the relatedness and relevance between the tasks performed.
3. Providing adaptive scaffolding on students' strategies when their task performances are below par.
4. Providing contextualized feedback as mixed-initiative conversations initiated by a scaffolding agent.

At the core of our adaptive scaffolding approach is a task model that provides a hierarchical breakdown of the primary OELE tasks into their component subtasks and observable actions. The top layer of the task model defines tasks common across a class of OELEs; the middle layer defines related subtasks specific to a particular OELE; and the lower levels map onto observable actions performed using the tools provided in the OELE. As a specific example, the CTSiM task model in Figure 2 breaks down the OELE tasks of information acquisition (IA), solution construction (SC) and solution assessment (SA) into CTSiM specific subtasks and actions. IA is linked to identifying and interpreting science and CT information by reading and searching through the science and CT resources and taking formative quizzes, SC covers using identified information to build conceptual and computational models of science topics, and SA covers verifying the models in their entirety or in parts by observing their simulations or comparing their simulations against expert simulations.

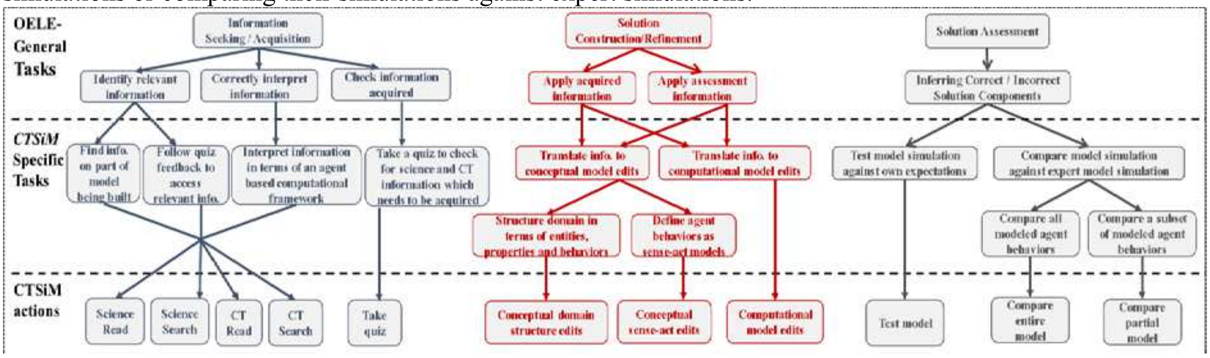

Figure 2. The CTSiM task model.

The task model does not specify any ordering or relations between sub-tasks or actions. These relations are represented by a strategy model that defines meaningful sequences of actions, subtasks, and tasks for accomplishing model building and learning goals specified in the OELE. For example, a set of action sequences that characterize features of individual actions (unary relations) and relationships between two or more action sequences (binary and higher-order relations) specify a 'strategy model' for CTSiM. In this work, we use a unary measure called '*effectiveness*', where effective actions move the learner closer to their corresponding task goal. For example, effective SC actions bring the learners' conceptual and computational models closer to a desired 'correct' model, and effective SA actions generate information about the correctness (and incorrectness) of individual agent behaviors modeled by the learner. Similarly, we adopt binary '*coherence*' metrics for defining effective strategies, where two temporally ordered actions or tasks ($x \rightarrow y$), i.e., x before y, exhibit the coherence relationship ($x \Rightarrow y$) if x and y share contexts, i.e., the context for y contains information contained in the context for x. The context for an action comprises the specifics of the action, such as the specific science or CT page read, the conceptual or computational components edited, or the agent behaviors compared. A general strategy definition can be hierarchically linked to more detailed versions that represent desired or suboptimal variants. By tracking student's activities, the system can compare strategy matches to desired versus suboptimal variants to estimate the student's proficiency versus need for scaffolding with respect to the strategy. The need for scaffolding is determined based on a combination of suboptimal action sequences and low modeling performance.

While several useful strategies can be defined using different combinations of tasks and actions from the CTSiM task model, we chose a set of five desired strategies (*S1–S5*) based on our previous observations of students' difficulties. We analyzed students' actions to detect deficiencies in these strategies.

S1. Desired: SC followed by coherent IA action (SC => Science Read)

Suboptimal: (ineffective SC → Science Read), i.e. ineffective SC action followed by an incoherent science read

S2. Desired: SA followed by coherent IA action (SA => Science Read)

Suboptimal: (effective SA detecting incorrect agent behaviors → Science Read), i.e. a SA action testing the model in parts and detecting incorrect agent behaviors followed by an incoherent science read action

S3. Desired: IA prior to solution construction or assessment strategy (Science Read => SC|SA)

Suboptimal: Lack of a Science Read action or an incoherent Science Read action before an effective SA action

S4. Desired: Test in parts strategy (Effective Compare action)

Suboptimal: ineffective Compare action

S5. Desired: Conceptual sense-act model building actions followed by coherent computational model building actions (Sense-act build => Computational build)

Suboptimal: incoherent (Sense-act build → Computational build) action sequence, or lack of the action sequence

S1, *S2*, and *S3*, link SC and SA actions to IA actions, implying the usefulness of seeking information about the part of the model they are building or assessing. *S4* describes a strategy for testing the model in parts to help isolate errors. *S5* pertains to SC, and how to effectively use multiple linked representations to build science models. When frequency counts for suboptimal uses of a strategy exceed a predetermined threshold (in the range of 2-5), scaffolds are triggered to provide feedback on use of the strategy. A local history of learners' conceptual and computational modeling skills are maintained by comparing different aspects of their models against the corresponding expert models to detect aspects of the modeling tasks they are struggling with. Separate 'missing' and 'extra' measures are maintained for different conceptual model components like agents, environment elements, properties, and behaviors chosen, as well as the sensed and acted-on properties specified for each agent behavior. Similarly, computational modeling skills are captured in terms of the number of missing and extra blocks, and whether all actions in a behavior occur under the right set of conditions. Students are scaffolded on their modeling tasks if their modeling skills do not improve between successive model assessments.

The task and strategy oriented scaffolds are all delivered in the form of a mixed-initiative conversational dialog initiated by Ms. Mendoza, and linked to students' recent actions and available information (e.g. simulation information or domain information). This conversation format engages students in a more authentic social interaction, and allows them to control the depth and direction of the conversation within the space provided by the dialogue and response choices. Our scaffolding approach helps students with a task or strategy only we detect that they are persistently facing problems, instead of correcting them every time we detect a problem. Scaffolds offer suggestions and reminders of good strategies and help point out possible sources of errors, but never provide '*bottom-out-hints*' by telling students exactly what to correct in their models.

Method

We conducted an experimental study with 98 students (average age = 11.5) from four 6th-grade sections in a Tennessee middle school. The science teachers assigned students from two sections to the control group (n = 46) which used a version of CTSiM without adaptive scaffolding, and students from the other two sections to the experimental group (n = 52) which received adaptive scaffolding.

The study was run daily over a span of three weeks during students' science periods (one hour daily for each section). All students worked on the same learning progression across two domains - Kinematics and Ecology. On Day 1, students took three paper-based tests that assessed their knowledge of (1) Kinematics, (2) Ecology, and (3) CT concepts. On day 2, students were introduced to agent based modeling concepts, and the whole class worked together on an introductory single-agent shape drawing activity using simple CT concepts like iterations. From Day 3, students worked individually. On days 3 and 4, they worked on generating growing and shrinking spiral shapes, which emphasized the relations between distance, speed, and acceleration. This activity was for practice only; students were allowed to seek help from their science teacher or from the research team if they had difficulties. From Day 5, students worked on the three primary modeling activities, and were not provided any individual help external to the system. On days 5 and 6, they worked on the first modeling activity, where they modeled the speed of a roller coaster (RC) car moving along different segments of a track. This required use of more complex CT constructs like conditionals. After Activity 1, students took paper-based Kinematics and CT post-tests on Day 7. On days 8-12, students progressed to modeling multiple agents with multiple behaviors in a fish tank system. In Activity 2, students built 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 was not modeled, the build-up of toxic fish waste caused the fish and the duckweed to gradually die off. In Activity 3, students addressed this problem by introducing micro-level entities, i.e., Nitrosomonas and Nitrobacter bacteria, which complete the waste cycle by converting the ammonia in the fish waste to nutrients (nitrates) for the duckweed. Students took their Ecology and CT-final post-tests on Day 13. Finally, on Day 14, they worked on a paper-based transfer activity where they started with a detailed textual description of a wolf-sheep-grass ecosystem and constructed conceptual and computational models of the ecosystem using modeling primitives specified in the question. Unlike the CTSiM environment, students did not have access to any of the online resources or tools.

All students' actions in the CTSiM system were logged to answer the following research questions:

1. Do the adaptive scaffolds make students' better science and CT learners?
2. Do the adaptive scaffolds improve students' modeling skills, and do these skills transfer beyond the CTSiM environment?
3. How do the adaptive scaffolds impact students' use of effective and suboptimal strategies?

We measured students' learning gains for kinematics, ecology, and CT by calculating their pre- and post-test scores. The Kinematics test assessed students' understanding of relations between speed, acceleration and distance. Students interpreted and generated speed-time and position-time graphs to explain motion in a constant acceleration field. The Ecology test focused on students' understanding of interdependence and balance in an ecosystem, and how a change in the population of one species in an ecosystem affects the other species. We assessed CT skills by asking students to predict program segment outputs, model scenarios and develop meaningful algorithms using CT constructs like conditionals, loops, and variables.

We assessed students' modeling skills using metrics similar to those used online for determining need for scaffolding (see Section 3). We computed the 'distance' between students' conceptual and computational models and the corresponding expert models in terms of the missing and extra model components, and normalizing it by the size of the expert model (i.e., the sum of the number of elements of each type of model component) to make the 'distance' measure independent of the size of the expert model (Basu et. al., 2014). We described students' modeling progress during an activity by calculating the model distances at each model revision and then characterized the 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. In order to study students' effective uses of strategies, we matched their logged action sequences with the definitions of desired strategies specified in Section 3. Suboptimal uses of the strategies were counted in terms of the frequencies of strategy feedback received by the experimental group and that would be received by the control group. The logged action sequences for students in the control group were matched against the suboptimal strategy variants, similar to the online strategy matching done for students in the experimental condition.

Findings

Science and CT learning gains and modeling proficiency

Our results show that students who received adaptive scaffolding had higher learning gains on both science and CT content. Table 1 summarizes students' pre-post learning gains for kinematics and ecology science content, and CT concepts and skills based on the CT test administered at the end of the study. Students in the experimental group had higher pre-test scores, hence we computed ANCOVAs comparing the gains between control and experimental conditions taking the pre-test scores as a covariate. Both groups had significant learning gains with medium to high effect sizes (Cohen's d), but the gains were higher in each case for students in the experimental group: kinematics gains ($F = 18.91$, $p < 0.0001$, $\eta_p^2 = 0.17$); ecology gains ($F = 52.29$, $p < 0.0001$, $\eta_p^2 = 0.36$); CT gains ($F = 40.69$, $p < 0.0001$, $\eta_p^2 = 0.31$). We also assessed students' performances on the first CT post-test at the end of kinematics unit, and found that students in the experimental group showed higher learning gains from the pre-test to the first post-test ($F = 18.16$, $p < 0.0001$, $\eta_p^2 = 0.16$) , and gained further from the second to the third CT post-test administered at the end of the ecology unit ($F = 18.85$, $p < 0.0001$, $\eta_p^2 = 0.17$).

To answer our 2nd research question, we compared the accuracy (distance to expert-model) of the final conceptual and computational models built by the control and experimental group of students. Figure 3 shows that students in the experimental condition built more accurate conceptual and computational models for all the activities (the final model distance scores were significantly lower) when compared to students in the control condition. Further breaking down the aggregate distance scores revealed that both missing and extra model constructs were significantly lower for the experimental condition for both conceptual and computational models. This implies that the experimental group's models included more model components from the expert model (lower missing score) and fewer redundant and incorrect components (lower extra score) than the control group's models.

In addition to building more accurate final models, the experimental group's progress towards the final conceptual model was significantly better than the control group as evidenced by three metrics: (1) higher percentage of effective (i.e., correct) conceptual edits in all three activities; (2) conceptual model accuracy improved with time in each activity, i.e., the slope for model distance over time was negative, whereas the distance slope for the control group was positive. (This was because the control group kept adding unnecessary elements to their models, and their conceptual models became more inaccurate in each activity as time progressed); and (3) modeling consistency was higher for the experimental group in the fish-micro unit. Also, the experimental group's computational model progressions within each unit were more consistent and improved more rapidly. Both conditions had negative computational model evolution slopes, i.e., their model accuracy improved over time in each of the activities. However, the rate of improvement was significantly higher for the experimental group in all the activities. We also analyzed students' performances on the transfer task and separately scored their conceptual and computational models of the wolf-sheep-grass ecosystem. We found that students in the experimental condition were better at applying their modeling skills and built more accurate conceptual ($p < 0.0001$, *Cohen's d* = 1.53) and computational ($p < 0.0001$, *Cohen's d* = 1.46) models compared to students in the control condition.

Table 1: Science and CT learning gains for students in the control and experimental conditions

		Pre	Post	Pre-to-post gains	Pre-to-post *p*-value	Pre-to-post Cohen's d
Kinematics (max = 45)	Control	12.52 (6.32)	15.55 (5.72)	3.03 (4.78)	<0.0001	0.55
	Experimental	16.65 (6.61)	22.38 (6.39)	5.72 (5.62)	<0.0001	0.88
Ecology (max = 39.5)	Control	7.40 (3.90)	16.19 (8.35)	8.78 (7.17)	<0.0001	1.35
	Experimental	9.39 (4.47)	27.91 (6.70)	18.53 (6.31)	<0.0001	3.25
CT (max = 60)	Control	16.49 (5.68)	22.53 (5.70)	6.04 (5.44)	<0.0001	1.06
	Experimental	22.72 (7.68)	32.24 (5.86)	9.52 (5.23)	<0.0001	1.39

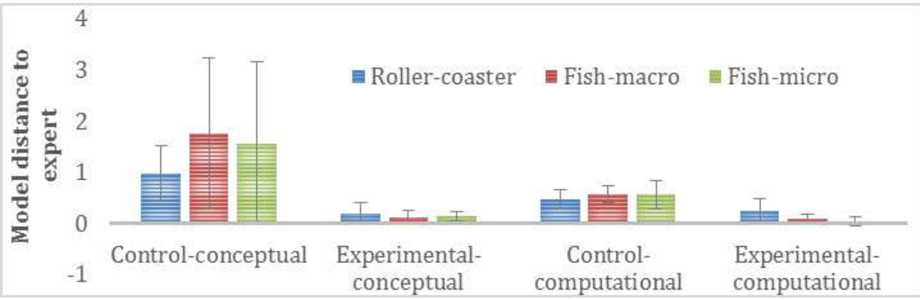

Figure 3. Modeling performance across conditions.

Effective and suboptimal uses of desired modeling strategies

To address our 3rd research question, we first computed the average number of times each of the five strategies was used effectively in each modeling activity, as well as the percentage of students who used the strategy effectively at least once in each activity (see Table 2). We note two general trends: (1) the fraction of students in the experimental group who used the strategies effectively was always greater than or equal to that in the control group, and (2) the average effective use of the strategies was also higher in the experimental group. While most of the differences had low to medium effect sizes (*Cohen's d* in the range of 0.2 to 0.7), the differences in use of the *Model-Build* strategy had much larger effect sizes in all three modeling activities (*Cohen's d* in the range of 1.36 to 1.75). Effective uses of this strategy were also strongly correlated with science learning ($p<0.0001$).

We also studied the effect of our adaptive scaffolds on students' suboptimal uses of strategies. Since the strategy oriented scaffolds were triggered based on the suboptimal strategy uses, we counted the feedback received in the experimental group and calculated the feedback that would be received by the control group. For each type of strategy feedback, Table 3 provides for each activity: (1) *n*, which represents the number of students who receive the feedback at least once in the activity, (2) *min-max*, which represents the lowest and highest number of times feedback is received by any student during the activity, and (3) *mean (s.d.)* represent the average number of times (and standard deviation) the feedback was received during the activity. We see that the experimental group students need significantly lower amount of strategy feedback than the control group would have needed, especially for the *Model-Build* strategy, the *test-in-parts* strategy, and the *IA-SC/SA* strategy. This implies that the adaptive scaffolds helped improve effective uses of the strategies, and reduced their suboptimal uses.

We also performed more fine grained analysis of effects of the scaffolds on effective uses of strategies by counting the number of effective uses before and after feedback instances. Our results show a general trend for students who needed scaffolding, their effective uses of strategies became more frequent as they received feedback for their suboptimal uses. For example, for *S4* (the test-in-parts strategy) in the fish-macro unit, 10 of the 52 experimental group students never received feedback on *S4* and made 0.8(1.5) effective uses of *S4* on an average. 15 students received feedback exactly once, and made an average of 2.0 (4.7) partial model comparisons before receiving feedback, which increased to 2.73(6.24) after receiving feedback. The other 27 students received feedback on *S4* two or more times; they used *S4* an average of 0.93(2.4) times before receiving any feedback, 1.93(4.2) times between the first and second feedback instances, and 4.7(7.43) times after receiving feedback twice.

Table 2: A comparison of the use of desired strategies across conditions (Note: *p<0.05, **p<0.01, ***p<0.001)

		RC		Fish-macro		Fish-micro	
Strategy		Fraction of students	Mean (s.d.)	Fraction of students	Mean (s.d.)	Fraction of students	Mean (s.d.)
S1. SC action followed by relevant science reads	C	37%	1.33 (2.99)	54%	2.43 (4.8)	70%	1.93 (2.05)
	E	63%	2.23 (4.71)	83%	4.75 (4.97)*	85%	3.4 (4.51)*
S2. SA actions followed by relevant science reads	C	4%	0.07 (0.33)	26%	0.76 (1.66)	26%	0.85 (9.31)
	E	38%	1.37 (2.69)**	44%	1.66 (2.29)*	44%	1.06 (0.24)
S3. Fraction of assessed behaviors that were read about before being assessed	C	80%	.73 (.42)	93%	.5 (.33)	83%	0.89 (0.27)
	E	92%	.86 (.28)	96%	.77 (.32)***	100%	0.96 (0.16)
S4. Number of partial-model comparisons	C	0%	na	48%	2.65 (5.79)	15%	0.57 (1.98)
	E	0%	na	58%	5.42 (7.16)*	19%	1.97 (3.22)*
S5. Fraction of sense-act properties removed or followed by a coherent computational edit	C	100%	0.67 (0.27)	100%	0.69 (0.31)	98%	0.59(0.31)
	E	100%	0.97 (0.1)***	100%	0.99 (0.03)***	100%	0.98 (0.06)***

Table 3: Comparing suboptimal uses of strategies in terms of feedback received or would be received (Note: *p<0.05,**p<0.005, ***p<0.0001)

		RC			Fish-macro			Fish-micro		
		n	Min-max	Mean(s.d.)	*n*	Min-max	Mean(s.d.)	*n*	Min-max	Mean(s.d.)
S1: SC-IA strategy	C	4	0-1	0.09 (0.28)	19	0-28	1.93 (4.46)	15	0-7	1.13 (1.98)
	E	3	0-1	0.06 (0.2)	22	0-4	0.69 (1.02)	8	0-1	0.15 (0.36)**
S2: SA-IA strategy	C	0	0	0(0)	0	0	0(0)	3	0-10	0.28 (1.5)
	E	0	0	0(0)	0	0	0(0)	0	0	0(0)
S3: IA-SC/SA strategy	C	16	0-57	8.43 (15.8)	41	0-62	18.8 (15.7)	8	0-14	1.11 (2.94)
	E	18	0-15	1.37 (3.11)**	19	0-14	1.81 (3.27)***	4	0-3	0.13 (0.52)*
S4: Test-in-parts strategy	C	0	0	0(0)	46	1-26	9.57 (6.77)	37	0-30	3.85 (5.27)
	E	0	0	0(0)	42	0-9	2.23 (2.13)***	23	0-6	0.83 (1.26)***
S5: Model-Build strategy	C	41	0-32	7.17 (6.19)	45	0-130	34.83 (28.87)	36	0-150	18.85 (26.95)
	E	32	0-8	1.79 (2.17)***	35	0-10	2.04 (2.43)***	30	0-10	1.33 (1.82)***

Finally, we also looked at how the task and strategy based scaffolds needed by the experimental group varied with time. We found that students needed a combination of task and strategy feedback in all the activities. In the initial RC activity, students received more task-oriented feedback than in the other two activities. In the more complex fish-macro activity, students needed more strategy feedback than in the RC activity, but less task feedback than in the RC activity, implying that the effects of the task feedback persisted across units. However, students found it challenging to manage and integrate the different tasks in a complex modeling activity involving a new domain. Finally, in the fish-micro activity, the task feedback received was further reduced, and the strategy feedback also decreased (to a smaller number than in the initial RC activity). This provides preliminary evidence that our scaffolding effects persisted, and, therefore, a fading effect occurred naturally as students worked across units. Further, the conceptual and computational models in the fish-micro activity were the most accurate of any activity, even though the students received less feedback in each category of scaffolds than in the earlier activities.

Discussion and conclusions

In this paper, we have presented a generalized adaptive scaffolding framework for OELEs and a specific example of its application in the CTSiM environment. Learning in CTSiM is based on an iterative model-test-refine cycle with a well-defined goal state. Representing a science topic using conceptual and computational representations helps students understand the science concepts underlying the topic, and offers them a chance to iteratively refine their understanding of the science concepts as they refine their models. Hence, our scaffolding approach does not provide students with the correct model at any point. Unlike several learning-by-modeling environments for science, which provide students with hints on incorrect relationships modeled, we combine students' modeling behavior and performance to determine their need for scaffolding. A study run with control (no adaptive scaffolding) and experimental (adaptive scaffolding) conditions demonstrates the effectiveness of our approach. The experimental group scored higher than the control group on science and CT assessments which were designed to test students' understanding of science processes and their reasoning and problem-solving skills as opposed to rote or inflexible knowledge. They also outperformed the control group in the ability to construct correct models, frequent use of effective strategies and infrequent use of suboptimal strategies. Further, we noticed a fading effect of our scaffolds - students in the experimental condition required less scaffolds across activities.

This work also contributes to the field of CT in K-12 education where few successful systems have been developed, assessments of students' learning are lacking, and scaffolds are limited to automatic assessments of students' computational artifacts based on the CT primitives and patterns contained. Our work provides an example of how CT principles can be operationalized and integrated with science curricula, and how scaffolds contextualized in science content can help students learn important CT concepts like sequences, loops, conditionals, and variables, and become more proficient in vital CT practices like decomposing complex tasks, testing and debugging, and abstracting and modularizing. Our results demonstrate significant correlations ($p<0.05$) between CT and science learning gains, as well as between important CT practices and science learning (Basu et. al., 2016). As future work, we plan to continue verifying our scaffolding framework with different OELEs and also in CTSiM with a more comprehensive set of strategies and global measures of students' performance and behaviors.

References

Azevedo, R. (2005). Using hypermedia as a metacognitive tool for enhancing student learning? The role of self-regulated learning. Educational Psychologist, 40(4):199–209.

Basu, S., Biswas, G. & Kinnebrew, J.S. (2016). Using multiple representations to simultaneously learn computational thinking and middle school science. In Thirtieth AAAI conference on Artificial Intelligence. Phoenix, Arizona, USA.

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, 61-100.

Duque, R., Bollen, L., Anjewierden, A., & Bravo, C. (2012). Automating the Analysis of Problem-solving Activities in Learning Environments: the Co-Lab Case Study. J. UCS, 18(10), 1279-1307.

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

Luckin, R. and du Boulay, B. (1999). Ecolab: The development and evaluation of a vygotskian design framework. International Journal of Artificial Intelligence in Education, 10(2):198–220.

Mitrovic, A.: 2011, 'Fifteen years of Constraint-Based Tutors: What we have achieved and where we are going'. User Modeling and User-Adapted Interaction.

Puntambekar, S. and H¨ubscher, R. (2005). Tools for scaffolding students in a complex learning environment: What have we gained and what have we missed? Educational Psychologist, 40(1):1–12.

Segedy, J.R., Kinnebrew, J.S., and Biswas, G. (2015). Using coherence analysis to characterize self-regulated learning behaviours in open-ended learning environments. *Journal of Learning Analytics, 2*(1), 13–48.

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

Acknowledgments

We thank participating teachers and students. This work funded by NSF Grant #1441542.