Making sense of models: How teachers use agent-based modeling to advance mechanistic reasoning

The MIT Faculty has made this article openly available. Please share how this access benefits you. Your story matters.

| Citation          | Ling, Hsiao et al. "Making sense of models: How teachers use agent-based modeling to advance mechanistic reasoning." British Journal of Educational Technology 50, 5 (July 2019): 2203-2216 © 2019 British Educational Research Association |
|-------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| As Published      | http://dx.doi.org/10.1111/bjet.12844                                                                                                                                                              |
| Publisher         | Wiley                                                                                                                                                                                              |
| Version           | Author’s final manuscript                                                                                                                                                                          |
| Citable link      | https://hdl.handle.net/1721.1/124342                                                                                                                                                              |
| Terms of Use      | Creative Commons Attribution-Noncommercial-Share Alike                                                                                                                                              |
| Detailed Terms    | http://creativecommons.org/licenses/by-nc-sa/4.0/                                                                                                                                                  |
Making sense of models:
How teachers use agent-based modeling to advance mechanistic reasoning

Ling Hsiao
Irene Lee
Eric Klopfer

Massachusetts Institute of Technology

Corresponding Author:
Ling Hsiao
700 Technology Square,
Building NE48
Cambridge, MA 02139
Email: lingh@mit.edu
Tel: 617-715-5779
eFax no: 601-767-7012
Abstract:

Computer modeling promotes mechanistic reasoning when learners build and analyze models of complex systems to explore causal mechanisms and use models to generate patterns. StarLogo Nova, an agent-based modeling environment, enables novice programmers to model a system's individual components and investigate its emergent, collective behavior. Through case analysis of teachers using StarLogo Nova, we demonstrate how agent-based modeling advances thinking about mechanisms generating phenomenon. Teachers who used simulation combined with the decoding of StarLogo Nova models utilized mechanistic reasoning to make sense of how and why complex phenomenon emerged.
Introduction

Modeling is a fundamental practice in the sciences that requires attending to underlying mechanisms of complex phenomena. Explaining how and why a phenomenon emerges entails knowing what entities and activities generate phenomenon. This important practice of analyzing for underlying mechanisms is called mechanistic reasoning (Machamer, Darden, & Craver, 2000; Russ, Scherr, Hammer, & Mikeska, 2008). Computer modeling environments are powerful tools that can develop students’ mechanistic reasoning while they build and examine computer models. Agent-based modeling advances student reasoning in the tradition of Logo computing environments. When students concretely instantiate representations of the natural world using dynamic, individual entities, they are using computer models as “objects-to-think-with” (Papert, 1980, p. 182). Students running agent-based models can observe patterns generated by entities’ activities and test whether their theories, encoded as agent interactions, generate aggregate-level behaviors of a complex system.

However, classroom lessons often incorporate pre-packaged science simulations (Perkins, Moore & Chasteen, 2014) that do not leverage the affordances of combining computer modeling with simulation. Presented as animations, these simulations may offer users the ability to change parameters and gather data but hide the mechanisms and rules governing the model’s behavior. Such simulations may reinforce knowledge of facts (e.g. who eats whom), but do not open up the ‘black box’ for students to examine why and how the outcome of a simulation occurs. Using only pre-packaged simulations limits student opportunities to develop mechanistic reasoning. In these cases students use models, but they are not engaged in modeling.

In this paper, we clarify how agent-based modeling (ABM) fosters mechanistic reasoning. StarLogo Nova (SLN) is a modeling environment developed within this constructivist tradition. SLN provides young learners the ability to build simulation models and analyze them for causal mechanisms. However, like other constructivist approaches showing promise in research, there are significant barriers to implementing ABM in schools (Wilensky, Brady, & Horn, 2014). Learning modeling in SLN requires scaffolding from a knowledgeable teacher. Hence, we address how teachers use the SLN environment to make sense of models and promote mechanistic reasoning in student thinking. Highlighting cases from our teacher professional development (PD) program, we investigate:

1. What patterns of StarLogo Nova usage support teachers’ mechanistic reasoning?
2. How do teachers attend to mechanisms using StarLogo Nova in student modeling activities?
Our study analyzes how teachers used features in the SLN environment during artifact-based interviews. We also examine observation data of teachers using SLN with students during modeling. Teachers’ investigative objectives in the interview are aligned with their chosen implementation approaches. These objectives and approaches demonstrate that SLN supports mechanistic reasoning when teachers leverage features that combine computer modeling and simulations use.

**Agent-based modeling: Investigating complex systems**

Models are purposeful representations of real systems (Grim et al., 2010). A modeler makes decisions on what to include in a model to study the system’s behaviors and its responses to change. In complex systems, aggregate-level patterns emerge from interactions across many individual entities at the local level (Resnick, 1994). ABM scientists program rules of behavior and interactions of individual entities, called *agents*, and repeatedly execute these rules to simulate how the system behaves over time. ABM is particularly effective for modeling emergent phenomenon (Railsback & Grimm, 2019).

In science classrooms, ABM can aid learners’ investigations of complex systems (Wilensky & Reisman, 2006). In modeling from an agentive perspective, students map real world entities and behaviors onto agents and then program them to interact with each other and with the environment mimicking natural interactions. The interactions generate patterns at the individual and population-levels for students to investigate. In SLN, students can thus create models that test provisional ideas of how the natural world works and run simulation experiments to generate and visualize outcomes. Pushed to explain their designs, students reason how and why a particular phenomenon happens. In the company of peers and an experienced teacher, this reasoning is further deepened during iterative refinements of modeling when students discuss the adequacy of a model (Passmore, Schwarz, & Mankowki, 2017). Deliberating on behaviors of agents that produce emergent phenomenon, students no longer simply watch simulations but think critically about the generators of phenomenon.

**Mechanistic reasoning framework**

Using mechanisms to explain population-level outcomes is a mode of causal reasoning often employed by students (Grotzer, Derbiszewska, & Solis, 2017). There is much ABM research on student explanations of complex systems using levels-based reasoning (Sengupta & Wilensky, 2009; Chi, Roscoe, Slotta, Roy, & Chase, 2012; Stroup & Wilensky, 2014). However, scholarship on how learners use mechanistic reasoning is sparse. Notably, Wilkerson, Shareff, Laina, and Gravel (2018) found that students who focused on entities’ movements and interactions developed progressively better mechanistic, explanatory models. Dickes, Sengupta, Farris, and Basu (2016) demonstrated that applying the lens of mechanistic reasoning to student explanations of agent-based models can trace students’ conceptual development of interdependence in ecology. Both studies appropriated Russ et al.’s (2008) framework on mechanistic reasoning to study students’ developing knowledge about causal mechanisms.
While not initially designed to study student inquiry about computer modeling tasks, Russ et al.’s (2008) discursive framework has been successfully applied to study student explanations of mechanisms in agent-based computer models. The framework articulates phenomena as composed of entities and their activities, and follows the changes of entity interactions from initial setup to intermediate stages of activities to termination conditions. Seven categories of explanations are ordered along a sequence of increasing sophistication. At the lowest Category 1 and 2, learners describe the target phenomenon of interest and setup conditions enabling mechanisms to run. The system’s entities, activities, and properties are identified in intermediate Categories 3-5. How the entities are structurally, spatially, or temporally organized in the mechanism is Category 6. The highest, Category 7, is chaining. When learners chain, they explicate the causal structure of phenomena to infer what happened to cause the current state of activities or predict what may happen next. They make claims about why a phenomenon occurs that provides evidence of mechanistic reasoning in their thinking. As such, the framework traces explanations of causal outcomes through agent behaviors and interactions. We applied a modified form of Russ et al.’s (2008) framework on both the interview and observation datasets, further described in the methods section.

*StarLogo Nova: Turtles, terrain, and interactions*

StarLogo Nova (SLN) is a web-based programming environment incorporating a powerful simulation engine and 3D renderer. Designed to engage young learners, it offers a visually appealing virtual world and color-coded user interface. SLN descends from Logo, a single-agent programming language widely used to support constructivist learning of mathematics and science (diSessa, Hammer, Sherin, & Kolpakowski, 1991). Unlike Logo, SLN allows users to control the behavior of thousands of agents at the same time. SLN’s block-based programming language is accessible to even elementary students, yet provides a ‘high ceiling’ for budding modelers to create more ambitious designs. Its interface consists of three distinct areas. The information window holds the title and description as well as the lineage of the project. Spaceland contains the 3D virtual world where agents are displayed. Spaceland has buttons enabling students to initialize and ‘run’ the model (simulation); sliders for setting values of input variables; graphs and data boxes for displaying output data; and a slider to control speed of simulation. Workspace includes ‘drawers’ of command blocks and ‘pages’ holding instructions for different agents. These instructions specify agent actions and/or update the agent’s state. Programs are constructed by snapping domain-general blocks together into sequences of instructions on pages. SLN was designed to enable fluid movements between the Workspace and Spaceland making multi-level connections possible.
Research Context and Methods:
Teachers implemented Project GUTS’ CS in Science, a curriculum that integrates computer science into school-day science classes through modeling and simulation. In the first module, students program SLN agents in a series of incremental ‘builds’ that culminate in creating a model of the spread of disease. Subsequent modules are positioned within earth, life, and physical science contexts and include a ‘base model’ that students can later decode (or analyze for abstractions and mechanisms), modify, and use to run simulation experiments.

Teachers with GUTS is a PD program that prepares educators to implement the CS in Science modules over the course of a year. Teachers attend a summer institute where they experience
the first module and one other module as adult learners. Follow-up workshops during the academic year focus on how to teach the modules. Our case data come from a cohort of teacher participants who work in public school districts in the southwestern region of the United States. They teach science in middle schools to students between the ages of 10 and 13 years old.

Six teachers in the cohort implemented the first module and offered a subsequent module during the second semester. We select cases from teachers who taught a second module because the latter modules focus less on acquiring basic SLN programming skills and more on understanding scientific phenomena through modeling. For this study, we compare three veteran teachers: Kevin, Dennis, and Maria. Each had at least seven years of teaching experience, taught SLN to most or all of their classes addressing students with a range of academic abilities, and managed class sizes that averaged 25 students, typical of most schools in this region.

We interviewed and observed participants. The mid-year artifact-based interview centered on teacher analysis of a computer model. It captured teacher interactions with the model and their explanations of how the model worked. Classroom observations were conducted by one or two researchers using a protocol that separated observers' concrete documentation of teacher and student actions from their impressions of classroom events. Each teacher was observed at least four times in the academic year, and for this study, lessons focused on modeling activities were analyzed.

Analysis

In both datasets, we coded for presence of mechanistic reasoning in teacher explanations and instructional talk using Russ and colleagues’ (2008) framework (see Table 1). However, this framework alone was insufficient in capturing several dimensions of our data. In the observation reports, we also coded inductively for teacher elicitations of student thinking (Cazden, 2001). In the interview, we applied a modified interactional analysis (Jordan & Henderson, 1995) to track on-screen actions in the SLN environment with teacher explanations of what was happening in the model. To compare teacher’s descriptive accounts and their aligned use patterns, we scored their overall explanations with a simplified rubric developed by Schwarz, Ke, Lee, and Rosenberg (2014) that corresponded to the Russ framework (2008). In the Schwartz rubric (2014), accounts that only identified mechanistic components, or what is in a phenomenon were ranked at a lower Level 1; this corresponded to the identification of entities, set up condition, and activities from the Russ framework (Categories 1-5). At Level 2, accounts explained how the mechanism is happening by describing the structural, spatial, and temporal locations of agent populations (Category 6). But case accounts that explain why the phenomenon is happening through chaining and descriptors of causal mechanisms (Category 7) scored the highest Level 3.
Table 1. Mechanistic reasoning framework mappings to teacher data

| Russ et al’s (2008) Categories | Description of Category: States or describes… | Examples of mapped mechanistic explanations from interview | Schwarz et al’s (2014) Levels |
|--------------------------------|---------------------------------------------|--------------------------------------------------------|--------------------------------|
| 1. Describes target phenomena (DTP) | Phenomenon of interest | “… This could be [about] wealth - Individuals that are somehow gaining power and earn money…” | |
| 2. Identifies setup conditions (SEC) | Conditions enabling mechanism to run | “I’m thinking that I’m going to vary the number of agents.” (Moves slider) | Level 1: accounts identifying non-mechanistic components, or “what” the phenomenon looks like |
| 3. Identifies entities (IE) | Agents that affect phenomenon outcomes | “Well, I think the agents represent the thing that dominates.” | |
| 4. Identifies activities (IA) | Actions and interactions of agent(s) | “Agents may not go in a pure straight line.” | |
| 5. Identifies properties of entities (IPE) | General properties of agents needed for mechanism to run | “So there are agents that are of different sizes.” | |
| 6. Identifies organization of entities (IOE) | How agents are organized, such as their arrangement structurally, spatially or temporally | “…where the largest crowds are” | Level 2: accounts explaining how the phenomenon is organized or proceeds over time |
| 7A. Uses chaining (IC) | Causal structure of the phenomena to make claims | “When the agents bumped into other agents, they enlarged.” | Level 3: accounts explaining why the phenomenon is happening through descriptions of causal mechanisms |
| 7B. Uses analogies (AN) | Phenomenon or mechanisms using analogies | “If you think of it, in economics, the richer are getting richer and the poor are getting invisible or worse.” | |
Findings

*Exploring the LegoMan model: Artifact-based interview*

The ‘artifact’ in the interview is based on a SLN model created in 2007 by a Project GUTS student investigating how one thing comes to dominate over time. This abstract model has relatively simple rules and a positive feedback dynamic in which initial growth leads to subsequent growth. The model can represent such phenomenon as ‘the rich getting richer.’ In the *LegoMan* model, small cubes are initially scattered randomly in Spaceland and start moving when the model is executed. Upon collision, larger cubes increase in size while smaller ones decrease. Over time, typically one large cube remains. Teachers can study the behavior of this system through interaction of two input variables: an initial number of agents and a maximum size of agents (See Figure 1). They can observe the outcome in the 3D world view, the accompanying population graph, and examine the code describing agent movements and interactions.

During initial exploration, teachers are tasked to “simply play with the model and see how it works.” In the following teacher cases below, we focus primarily on these first three minutes where teachers demonstrate their unique approaches in using SLN features to understand the LegoMan model. While it may be difficult for a novice to analyze all of the different mechanisms in three minutes, we believe it is enough time to figure out at least one mechanism in a simple model. In their free-form investigations without interviewer prompts, teachers’ interactions with the SLN model reveal their investigative objectives. These differing approaches eventually led to different mechanistic explanations.

In the full 20-minutes interview, teachers were asked to examine and interpret specific components of the model. For example, interviewers asked teachers to identify agents and activities, read and share their understandings of mechanisms in the code, and evaluate the model in terms of its abstractions and assumptions about the phenomenon. Each teacher successfully decoded the setup and runtime procedures including the mechanisms governing agent interactions under Workspace. All three also interpreted the LegoMan model as modeling the phenomenon of something becoming dominant over time, such as “financial growth” or “consumption.”

*Kevin: Simulation only (Figure 2)*
Kevin initially inquired whether “playing with the model included viewing the code [sic].” Despite assurances that he could interact with any tool feature, Kevin chose to only interact within the Spaceland simulation area. His interaction was *simulation-only*.

Kevin was primarily interested in how the model system behaved under different conditions. After clicking setup and forever buttons to initiate and execute the program, Kevin identified entities (IE) and activities (IA), remarking that he was “watching SpaceLand” to look for “patterns.” He noted changes in properties of entities (IPE): “Some things are getting smaller and some things are getting bigger…I want to do it [run it] again.” On his second try, he shared: “I’m noticing when they seem to get bigger but not necessarily in proportion of what they’re colliding with. It seems to be on the next financial growth.” Kevin’s observations about changes as a result of collisions hinted at causality (IC), and he interpreted the interaction as analogous to gaining wealth (AN).

Kevin ran four experiments setting initial conditions with sliders and watching the simulation in Spaceland. Each time, he varied the number of agents and agents’ maximum size before making observations. For example, setting the max size of agents to 50, then initializing and running the model, he stated: “Increase number of agents (IOE), and that’s what I think ... population decreases rapidly (IOE).” Kevin was exploring the models’ behaviors under various setup conditions.

Applying the Schwarz rubric (2014), Kevin’s model-based explanations at first focused on non-mechanistic descriptions such as appearance (Level 1-What) and later addressed how a phenomenon happens when he described the correlations between input variables and experiment outcomes (Level 2-How). While his analysis of Spaceland alluded to some chaining and analogies of model behaviors, his explorations within simulation-only interaction did not yield an overall understanding of explanatory processes or why the phenomenon occurred.
Dennis: Simulation and decoding - confirming in simulation what he saw in code (Figure 3)
In contrast, Dennis interacted within the Spaceland window and the Workspace area. His type of interaction was *simulation and decoding*, and his reasoning strategy entailed using simulation runs to confirm what he understood about the underlying mechanisms in code. Initially, Dennis ran the model and quickly identified a property of the entities (IPE): “They seem to get larger.” He also observed that some agents changed sizes (IPE) contemporaneously “when they bump into each other and some of them seemed to combine.” He adjusted a slider to increase the simulation speed and confirmed his initial observation: “They’re definitely getting bigger when they collide with each other.” Like Kevin, Dennis ran an experiment to identify changes in agent properties. Next, Dennis glanced at the graph and remarked: “The population is dropping (IOE).”

But unlike Kevin, Dennis then ventured into Workspace to read the agent’s code. He shared a new insight about agent properties (IPE): “The collision block says - I didn’t notice that - some things we would get smaller. I want to run it again and see from the beginning.” Dennis attended to Spaceland and stated: “Let’s see if any of them get smaller when they – oh, yeah, cool. So, some of them, if they’re a certain size, they get larger and if they’re smaller than that then they get smaller.” Dennis ran the model a second time to confirm the agent behavior he just decoded but missed in his first observation of the simulation.
Dennis’s combined use of simulation and decoding reflected a desire to know how the model system was constructed. Only when Dennis decoded did he yield causal mechanistic explanations, noting that after collision, if agents were small “then they get smaller” (Level 3-Why). He deduced the mechanism for changing in size by reading the code before checking the behavior of agents in the simulation.

Figure 3. Dennis’s interactions in StarLogo Nova and think-aloud statements

Maria: Simulation and decoding - confirming in code what she saw in the simulation (Figure 4)

Like Dennis, Maria’s interaction combined simulation and decoding. However, Maria ran the simulation twice without manipulating sliders before scrolling down to see the Workspace area. She started by identifying entities (IE), their properties (IPE), and their activities (IA) in Spaceland: “The agents are being measured in the drawing. Oh, one turned blue! That’s just different from the rest and some seem to float which makes me wonder what’s going on…I’m going to reset it…” After running the program a second time, Maria made causal claims (IC) while watching the simulation: “Oh, when I hit maximum size, that’s when I change color… It looks like one block was consuming another which is why they’re getting bigger.” She used the analogy of “being consumed” (AN) to help her make sense of the interactions over time.

Turning to Workspace next, Maria began by assessing the model: “It’s not very complex world, it’s very simplistic.” She then decoded a causal mechanism in the program and confirmed the etiology of what she had observed previously: “When I hit maximum size, that’s when I change color - that’s what the code tells me. If not, I stay white...If my size is greater than the size of colliddee [other agent], then I increase my size by 0.5 and then move forward one. If it’s not, I’m going to decrease by 0.5, and move backwards.” Maria observed simulated mechanisms in action and then confirmed those observations by reading the causal mechanisms in the code. She connected how the model behaved and how it was constructed. Her descriptive accounts of
how “one block was consuming another” addressed how a phenomenon happens (Level 2-How) and her analysis of ‘IF-THEN’ conditional statements explained why the phenomenon occurred (Level 3-Why).

Figure 4. Maria’s interactions in StarLogo Nova and think-aloud statements

Approaches to teaching modeling: Classroom observations
We observed that teachers’ approaches to modeling in subsequent modules reflected their investigative objectives in the artifact-based interview. Kevin, who focused on the model’s behaviors, conducted more lessons using models as demonstrations. He had students analyze model system behaviors. Dennis, who concentrated on how the model was constructed, spent considerable in-class time coding and modifying base models. Maria drew connections between model system behaviors and constructions; she had students research consumer-producer relationships and then challenged them to build their own models of the phenomena. Across cases, teachers capitalized on the affordances of the SLN environment to promote students’ mechanistic thinking. We found evidence of mechanistic reasoning in instructional talk, as exemplified through one example in each teacher case below.

Kevin’s demos
Kevin’s instruction prioritized student analysis of ready-made SLN models. Although his students modified some model components, comparatively, his class spent less time programming than Dennis or Maria’s class. Using existing models as demonstrations, Kevin taught his class to carefully examine properties of a model through decoding. He had students describe model system behaviors and later examine the codes.
Kevin displayed a model depicting heat causing Earth’s temperature to rise. His students had just learned about the ice-albedo feedback and the impact of greenhouse gases on climate change in their regular science unit. A sun generated small yellow agent balls (solar energy) that fell to the earth’s surface in Spaceland. Upon contact, some changed to white balls (reflection) while others into infrared ‘heat’ (radiation), illustrating the absorption of solar energy and later, the release of energy.

Kevin initially ran the model without much explanation, asking students to share observations of agent behaviors and what they think the agents represented. One girl guessed “red is heat.” Kevin inquired what the small yellow balls might be. Many students immediately answered: “carbon dioxide.” A boy argued that the slow down and up movements portrayed floating gas molecules. Students began to describe the agents as gases moving through earth’s atmosphere.

Thereupon, Kevin changed to code view in Workspace. He showed the procedure for the small yellow ball that “is creating energy” in the Solar Energy Tab and decoded aloud:

Kevin: If z value is greater than 0, then it goes down. If it is less than 60, it is reflected...If the random number is less than 20, it goes up 0.1 step. How quickly does it move in nature? The speed of light. There is no mention of gases. This model does not have greenhouse gases.

Kevin guided student inspection of coded mechanisms governing the yellow agent’s interactions with the simulated terrain. He emphasized how the agent’s movements were programmed to “the speed of light” despite slow-moving in the simulation. Essentially, Kevin’s demonstration of decoding identified the agent’s properties as solar energy, not gases. Clarifying what entities and properties were expressed in the code redirected student attention to the mechanisms of rising temperature and thus promoted student mechanistic reasoning.

**Dennis’s modified models**

Dennis devoted more class days to programming, offering opportunities to ‘see’ how chemical equations were constructed. As pairs coded, he helped debug errors in student models and in the process, cleared misunderstandings about the physical mechanisms taking place in chemical reactions. In one session, students were modifying a basic SLN model of silver nitrate and copper reaction by coding new ‘breeds’ of agents to simulate the formation of silver and aqueous copper nitrate products. Dennis did not receive materials in time to demo the wet lab before modeling. Instead, he focused on the chemical equation to represent the phenomenon since his students had just learned about the conservation of mass in balanced equations. Hence, while students knew what products would form based on the balanced equation in this replacement reaction, they had limited knowledge of what the reaction would look like or how it worked.
Midway, two boys alerted Dennis that their model was successfully “creating silver.” Dennis glanced at Spaceland and asked whether they had coded silver agents ‘green’ as he couldn’t see the agents in the simulation. As the pair checked agent traits in Workspace, he examined their procedures:

Dennis: You are deleting it. Think about where you have [the code for silver] nested. You are creating it and deleting it, creating it and deleting it.

Pair fixes and runs the program. Boy zooms Spaceland into the copper rod.

Dennis: You have your silver walking around? You need to let it sit there…It is supposed to replace the copper rod.

Shifting attention from Spaceland to Workspace, Dennis helped the boys debug the order of the programming blocks. More importantly, as the pair tested their simulation, he pointed out that the physical mechanism was modeled incorrectly because silver should form on the copper rod after a replacement reaction. In troubleshooting errors, Dennis also traced the productive changes that should take place after a reactive ‘collision’ between different reagents. This mechanistic description could prompt chaining in student thinking when students make sense of cause and effects during a reactive interaction. As such, Dennis attended to mechanistic reasoning when students constructed codes to simulate chemical reactions.

Maria’s created models
Unlike Kevin and Dennis, Maria had students build their own base models from scratch. Students researched and then coded a model of consumers and producers, such as a rabbit and grass model representing an ecosystem of two trophic levels. She directed student attention to the relationship between consumers and producers, and this relationship was key in deciding what to abstract and model. For example, after programming their base models, Maria facilitated a discussion comparing students’ own models of ecosystems to those in the real world. The contrast accentuated missing components and mechanisms underlying student models. In her science classroom, Maria had just completed a unit on decomposition where students conducted soil experiments in a cup, and she referred to this lab experience to highlight the contrasts.

Maria asked her class whether their simple models of ecosystems were “perfect.” Many did not and shared examples like agents “don’t get sick.” Another girl remarked that her model was missing “a top to the food chain.” Maria then referred to a model built by two boys depicting sharks attacking humans:

Maria: How do the humans get energy? There’s nothing for them to consume?

Maria points and asks how the ecosystem is like the cups of soil in their classroom.

Maria: What gets put into soil?
Several students: Rocks. Minerals. Dirt.
Maria: Other things besides what comes from rocks?
Student1: Decomposed plants and animals.

Maria: So we are not showing in our ecosystem how the animals can decompose? In our rabbits and grass model, what might happen to the grass if there’s a decomposed rabbit?
Student2: Soil might become more fertile.

In this exchange, there was presence of strong mechanistic reasoning in Maria’s instruction. Students had learned in lab experiments that decomposition recycles consumer organic compounds back into the inorganic soil to provide essential nutrients for future producer growth. Linking students’ modeling experience back to their prior lab experiments, Maria pushed students to consider missing mechanisms in their representations. Student responses like “soil might become more fertile” demonstrated forward chaining thinking because students used causal structure to conjecture what should happen next in their models. The discussion sparked mechanistic reasoning and provided the basis for future modeling sessions where students added a third trophic level to their initial simple model.

Discussion
Teachers’ investigative objectives when faced with a novel model paralleled their approaches to teaching modeling to their students. Kevin noticed model system behaviors in the interview and tended to hone in on student observations of model behaviors during demonstrations of SLN models. Dennis may have started with a brief observation of the running model but prioritized searching for mechanisms in the code. He sought how the model was constructed and moved from Workspace to Spaceland to verify his decoding. Similarly, Dennis emphasized in instruction coding and debugging in service of modifying a model. In contrast, Maria’s instruction centered on models as imperfect abstractions of the real world. In the interview, she commented on the model’s high level of abstraction before carefully observing the model’s behavior and validating those observations in the code. She connected model behavior and construction to make sense of the abstracted mechanisms. In the classroom, Maria also engages her students in thinking about abstractions embedded in models and how missing mechanisms may limit a model’s predictive accuracy.

Comparing across cases, we noted that in the artifact-based interview, different patterns of SLN usage yielded different mechanistic understandings. Although all teacher think-alouds showed
some presence of mechanistic reasoning, their overall explanations about the model differed. Observations of the model running in Spaceland (Level 1 explanations as scored in the Schwartz rubric (2014)) focused on what the phenomenon looks like and Level 2 explanations addressing how the phenomenon proceeded over time. Only when observations of Spaceland were combined with examining code did Level 3 explanations emerge, ones that use mechanistic descriptions to explain why phenomenon occurred.

Likewise, we saw presence of mechanistic reasoning in instructional talk when teachers used simulation combined with coding. Regardless of the approach teachers used, their instruction attended to critical evaluation of simulations. Their strategies included decoding of the SLN model to promote student mechanistic reasoning. As such, our study demonstrated that teachers’ sense-making of computer models were enabled through the StarLogo Nova environment. In particular, SLN features and tools provided the ability to:

- Inspect code for mechanism to understand how a specific process was abstracted and encoded.
- Move between Spaceland and Workspace to make connections between individual agent behaviors and larger simulated population behaviors
- Move between Spaceland and Graph to confirm temporary patterns at the aggregate-level.
- Adjust the speed of simulation to enable slowing or speeding up the simulation for targeted viewing of individual or aggregate behaviors.
- Set variables and run experiments to understand the complex system’s behavior across multiple parameters and generate a landscape of outcomes.

There were limitations in our analysis. We observed that Russ et al.’s (2008) mechanistic reasoning framework did not fully capture instances of teacher thinking about agent-based models. One issue is related to the nature of complex systems. Although Russ and colleagues’ characterization of mechanistic thinking in modern scientific practices align with current understandings of complex systems, we note the danger of coding for mechanisms from a “clockwork” orientation, a perspective that may not fully capture reasoning about dynamic processes of emergence (Yoon, Got, & Park, 2018).

As use of simulations become commonplace in science classrooms, we advise against using simulation-only experiences consisting of ‘active-in-behavior’ exercises. Attending to the decoding of models, teachers can support students’ active thinking about the physical mechanisms governing complex systems in the natural world.
Acknowledgments
We thank Susan Gibbs, Patricia Meyer, and Mary Sagartz for their assistance in this study. This project is supported under NSF Awards DRL-1503383 & DRL-1639069.

Statements on ethics, open data, and conflict of interest
The Committee on the Use of Humans as Experimental Subjects at Massachusetts Institute of Technology approved this research (approval #1604545318). Due to ethical reasons protecting identities of teachers, this study’s data cannot be accessed. All authors have no conflict of interest in the work reported here.
References

Cazden, C. B. (1988). *Classroom discourse: The language of teaching and learning.* doi:10.1017/s0047404500014676

Chi, M. T., Roscoe, R. D., Slotta, J. D., Roy, M., & Chase, C. C. (2012). Misconceived causal explanations for emergent processes. *Cognitive science, 36*(1), 1-61. doi:10.1111/j.1551-6709.2011.01207.x

Dickes, A. C., Sengupta, P., Farris, A. V., & Basu, S. (2016). Development of Mechanistic Reasoning and Multilevel Explanations of Ecology in Third Grade Using Agent-Based Models. *Science Education, 100*(4), 734-776. doi: 10.1002/sce.21217

diSessa, A., Hammer, D., Sherin, B., & Kolpakowski, T. (1991). Inventing graphing: Children’s metarepresentational expertise. *Journal of Mathematical Behavior, 10*(2), 117–160.
Grimm, V., Berger, U., DeAngelis, D. L., Polhill, J. G., Giske, J., & Railsback, S. F. (2010). The ODD protocol: a review and first update. Ecological modelling, 221(23), 2760-2768. doi: 10.1016/j.ecolmodel.2010.08.019

Grotzer, T. A., Derbiszewksa, K., & Solis, S. L. (2017). Leveraging fourth and sixth graders’ experiences to reveal understanding of the forms and features of distributed causality. Cognition and Instruction, 35(1), 55-87. doi:10.1080/07370008.2016.1251808

Jordan, B., & Henderson, A. (1995). Interaction analysis: Foundations and practice. The journal of the learning sciences, 4(1), 39-103. doi:10.1207/s15327809jls0401_2

Machamer, P., Darden, L., & Craver, C. F. (2000). Thinking about mechanisms. Philosophy of science, 67(1), 1-25. doi:10.1086/392759

Papert, S. (1980). Mindstorms: Children, computers, and powerful ideas. New York: Basic Books, Inc. doi: 10.1037/h0098915

Passmore, C., Schwarz, C. V., & Mankowski, J. (2017). In Schwarz, C. V., Passmore, C., & Reiser, B. J. (Eds.), Helping students make sense of the world using Next Generation Science and Engineering Practices. Arlington, VA: NSTA Press. doi: 10.2505/9781938946042

Perkins, K. K., Moore, E. B., & Chasteen, S. V. (2014, July). Examining the use of PhET interactive simulations in US college and high school classrooms. In Proceedings of the 2014 Physics Education Research Conference (Minneapolis, MN, USA (pp. 207-210). doi: 10.1119/perc.2014.pr.048

Railsback, S. F., & Grimm, V. (2019). Agent-based and individual-based modeling: A practical introduction. Princeton, NJ: Princeton University press. doi: 10.1007/springerreference_60185

Resnick, M. (1994). Turtles, termites, and traffic jams: Explorations in massively parallel microworlds. Cambridge, MA: MIT Press. doi: 10.1016/S0898-1221(97)90147-4

Russ, R. S., Scherr, R. E., Hammer, D., & Mikeska, J. (2008). Recognizing mechanistic reasoning in student scientific inquiry: A framework for discourse analysis developed from philosophy of science. Science Education, 92(3), 499-525. doi:10.1002/sce.20264

Schwarz, Christine & Ke, Li & Lee, May & Rosenberg, Joshua. (2014). Developing mechanistic explanations of phenomena: Case studies of two fifth grade students’ epistemologies in practice over time. Proceedings of International Conference of the Learning Sciences, ICLS. 1.
Sengupta, P., & Wilensky, U. (2009). Learning electricity with NIELS: Thinking with electrons and thinking in levels. International Journal of Computers for Mathematical Learning, 14(1), 21-50. doi:10.1007/s10758-009-9144-z

Stroup, W. M., & Wilensky, U. (2014). On the embedded complementarity of agent-based and aggregate reasoning in students’ developing understanding of dynamic systems. Technology, Knowledge and Learning, 19(1-2), 19-52. doi:10.1007/s10758-014-9218-4

Wilensky, U., Brady, C. E., & Horn, M. S. (2014). Fostering computational literacy in science classrooms. Communications of the ACM, 57(8), 24-28. doi:10.1145/2633031

Wilensky, U., & Reisman, K. (2006). Thinking like a wolf, a sheep, or a firefly: Learning biology through constructing and testing computational theories—an embodied modeling approach. Cognition and instruction, 24(2), 171-209. doi: 10.1207/s1532690xci2402_1

Wilkerson, M. H., Shareff, R., Laina, V., & Gravel, B. (2018). Epistemic gameplay and discovery in computational model-based inquiry activities. Instructional Science, 46(1), 35-60. doi:10.1007/s11251-017-9430-4

Yoon, S. A., Goh, S. E., & Park, M. (2018). Teaching and learning about complex systems in K–12 science education: A review of empirical studies 1995–2015. Review of Educational Research, 88(2), 285-325. doi:10.3102/0034654317746090