J Sci Educ Technol (2015) 24:265–286
DOI 10.1007/s10956-014-9506-8

Sandboxes for Model-Based Inquiry

Corey Brady · Nathan Holbert · Firat Soylu · Michael Novak · Uri Wilensky

Published online: 13 July 2014

Abstract In this article, we introduce a class of constructionist learning environments that we call *Emergent Systems Sandboxes* (*ESSs*), which have served as a centerpiece of our recent work in developing curriculum to support scalable model-based learning in classroom settings. ESSs are a carefully specified form of virtual construction environment that support students in creating, exploring, and sharing computational models of dynamic systems that exhibit emergent phenomena. They provide learners with "entity"-level construction primitives that reflect an underlying scientific model. These primitives can be directly "painted" into a sandbox space, where they can then be combined, arranged, and manipulated to construct complex systems and explore the emergent properties of those systems. We argue that ESSs offer a means of addressing some of the key barriers to adopting rich, constructionist model-based inquiry approaches in science classrooms at scale. Situating the ESS in a large-scale science modeling curriculum we are implementing across the USA, we describe how the unique "entity-level" primitive design of an ESS facilitates knowledge system refinement at both an individual and social level, we describe how it supports flexible modeling practices by providing both continuous and discrete modes of executability, and we illustrate how it offers students a variety of opportunities for validating their qualitative understandings of emergent systems as they develop.

Keywords Constructionism · Design · Agent-based modeling · Scalability

C. Brady (✉) · N. Holbert · F. Soylu · M. Novak · U. Wilensky
Northwestern University, 2120 Campus Drive, Evanston, IL 60208, USA
e-mail: cbrady@northwestern.edu

Introduction

In this article, we describe and illustrate a novel constructionist (Kafai 2006; Papert and Harel 1991) approach to model-based inquiry (Buckley et al. 2004; Lehrer and Schauble 2006; Windschitl et al. 2008). Specifically, we introduce a class of learning environments that we call *Emergent Systems Sandboxes* (*ESSs*), which have served as a centerpiece of our recent work in developing curriculum to support scalable model-based learning in classroom settings. ESS environments are a carefully specified form of construction environments that support students in creating, exploring, and sharing virtual models of dynamic systems that exhibit emergent phenomena. They differ from computational tools used in other model-based design work, both in terms of their affordances for construction and the ways we use them to structure inquiry-driven explorations for students. We present our ESS construct as a response to the Special Issue's question, "How can technology transform teaching and learning as students develop and use models?" and we argue that ESSs offer a means of addressing some of the key barriers to introducing and pursuing rich model-based inquiry approaches in science classrooms at scale.

We begin by indicating the research context for our design work. We then provide a preliminary definition of an ESS focused on the *construction primitives* it offers to the learner and the relation of these primitives to core disciplinary knowledge structures; the means it offers users to *run* their constructions in various ways; and its use of *saving state* to support individual and social exploration of the behaviors of systems constructed by learners from its primitives. Next, we describe our theoretical framework, which attends to the individual and social dimensions of our design of the ESS. We then proceed to offer a more elaborated description of the ESS construct and of the specific ESS we developed for our Particulate Nature of Matter (PNoM) unit. Finally, rooted in this description, we discuss three features of ESS environments and of the classroom interactions they foster, which have emerged in the course of our iterative design and implementation work with this unit.

Research Context

Our development of the ESS design construct has emerged in the context of our work on the ModelSim project (NSF# DRL-1020101). A core objective of this project is to investigate the *scalability* of three innovative approaches to science learning supported by the NetLogo environment (Wilensky 1999a). These approaches are as follows: (1) model-based inquiry with agent-based modeling (ABM), in which scientific phenomena are viewed as emergent, system-level behaviors and studied through computational simulations; (2) participatory simulations (PartSims), in which classroom groups engage in systems simulations by taking on the role of agents in the system to experience emergent phenomena firsthand; and (3) bifocal modeling, in which students connect virtual agent-based models with the physical world by taking in data streams from sensors and by generating physical behaviors through motors and other outputs. Each of these modalities has been developed over a long history of design research. The ModelSim project investigates their systematic combination and use in extended curricular units (2 weeks, 10–12 of hours of class time) treating core science topics at the high school level: Population Dynamics; Evolution; Electricity; and the PNoM. The project aims to refine and study both the technological and pedagogical supports for these three modalities from perspectives of scalability.

In this article, we focus on work connected with the PNoM unit of the project and especially with the first week of that 2-week unit. We do this in order to describe the ESS construct in a particular curricular context that illustrates key aspects of the ModelSim project's conception of scalability. Specifically, we identify two dimensions of scalability that we have considered in our design.

The first dimension of scalability deals with the *size* of implementations, as a matter of numbers. In this sense, one of the challenges of scaling innovative curriculum involves replicating essential features of learning environments developed in laboratory studies in more complex classroom settings. An objective of ModelSim along this dimension is to support widespread, effective use of construction tasks that are critical for students to develop deep, mechanistic understandings of the emergent systems explored in each of our units.

An extensive history of design research on constructionist curriculum in the domain of Chemistry and the Ideal Gas Laws provides a foundation for our PNoM work. Early work with an agent-based environment to explore the PNoM, called GasLab (Wilensky 1999b, 2003) showed the learning potential of engaging students in developing their own computational simulations of phenomena related to the behavior of ideal gases. Here, the process of constructing a computer model became a process of debugging not only one's code but also one's conceptions of particle behaviors and the implications of these behaviors on aggregate phenomena such as temperature, pressure, and volume (Wilensky 2003).

Later work associated with the Connected Chemistry curriculum (Wilensky et al. 2004; Levy et al. 2006; Stieff and Wilensky 2003) sought to scale the GasLab research in our first, numeric sense. It provided a more structured sequence of investigations that were experienced by a much larger group of students across a variety of school and classroom settings, as a part of the Modeling Across the Curriculum project (Gobert et al. 2003; Levy and Wilensky 2009a, b; Levy et al. 2006; Stieff and Wilensky 2003). Although Connected Chemistry was quite successful, the roles of *construction* and *programming* in that curriculum were substantially reduced from the original GasLab studies. Nevertheless, an extension activity within Connected Chemistry did provide an open, exploratory environment, called a "Particle Sandbox," which in fact served as the initial inspiration for our PNoM work. Our ModelSim design objectives with respect to this first dimension of scaling are thus to build on GasLab and Connected Chemistry, continuing the pursuit of ways to engage students in open-ended construction tasks and to assess the viability of these tasks in learning environments beyond the controlled settings of researcher-supported laboratory studies.

A second dimension of scalability has also proved important to our learning design work in the ModelSim project: scaling from the individual learner to the classroom group. An innovation is scalable in this sense if it makes effective use of the *social* resources and structures of classrooms, providing a basis for the construction of shared collective understanding. In particular, such classroom-level scalability can be achieved when students experience each others' work as comprehensible and relevant to their own, and as building toward shared or complementary goals. When this occurs, students act as an engaged and authentic, critical audience for each other's work. Such settings also encourage an active facilitation role for the teacher, to support students in making the most of the findings of their classmates and of the group as a whole.

The ESS contribution we describe in this paper is a type of learning environment that supports this dual sense of scale for model-based inquiry. Because the success of the ModelSim project depends upon our addressing scale at both of these levels, the affordances of ESSs will be measured by their ability to facilitate this scaling. For instance, in the current academic year we are implementing our curricular units in approximately 100 classrooms, and as will be clear in the discussion below, effective small-group and whole-class interactions are essential to the successful functioning of the units. Thus, two scalability-related design requirements for the ESS are:

1. It must permit students to engage deeply in individual, personally meaningful model construction and model-debugging processes, and
2. It must permit students to share their constructions with classmates and to benefit from interacting with and discussing these artifacts, as an integral component of their learning *during* the construction process.

These two requirements stem from the numeric and classroom-level dimensions of scaling, respectively. And though they can be considered independently, we will show how the ESS serves to connect these two dimensions. Indeed, the ESS works to *bridge* individual and social aspects of modeling, using each as a means of addressing challenges associated with the other.

A Preliminary Definition of an ESS

With the context and design challenges of ModelSim in mind, we can offer a preliminary definition of an ESS. An ESS is first and foremost an *agent-based* computational modeling environment for creating and exploring emergent complex systems. In an agent-based approach to emergence, rather than describing and measuring phenomena at the aggregate level (using differential equations or a systems dynamics model with "stocks" and "flows"), one instead conceives of these aggregate-level phenomena as emerging from the interactions of many individual, autonomous "agents." ABM, then, is about attempting to identify the individual agents of a system along with the behaviors and rules that these agents follow that will lead to the emergence of target phenomena at the aggregate level. An agent-based perspective has been shown to be a powerful approach for explaining and understanding phenomena across a wide range of domains, including the natural sciences (Abrahamson and Wilensky 2004; Blikstein and Wilensky 2004; Levy et al. 2006; Sengupta and Wilensky 2005; Wilkerson-Jerde and Wilensky 2010).

With this foundation in ABM, an ESS has three key properties:

1. It offers its construction primitives at the "entity" level, where we define entities as agents-with-fixed-behaviors, and where these fixed behaviors are governed by a core scientific model that underlies the construction environment.
2. It allows flexible execution of constructions, enabling users to run them both continuously as they build and discretely as coherent runs that produce outcomes.
3. It supports saving and sharing of states of constructions, facilitating iterative experimentation, and sharing of in-process artifacts with peers.

In later sections of the article, we will unpack this definition further, showing how we applied the construct to create the particular ESS we used in the introductory week of our PNoM unit and then illustrating the affordances of that ESSs through the work of students and classroom groups who engaged with it to build and explore models of diffusion.

Theoretical Framework

A wide range of research in mathematics and science education has indicated the power of modeling activities, both in illuminating student thinking and in producing conceptual change. Specifically, researchers have shown that model-based learning approaches can support content mastery (Stewart et al. 2005), competence, and fluency with disciplinary practices such as argumentation (Passmore and Svoboda 2012) and assimilation of meta-knowledge required to engage in these practices in appropriate ways as a part of authentic inquiry (Schwarz et al. 2009). Moreover, the Next Generation Science Standards (NGSS) situate developing and using models as one of the eight core scientific practices. In particular, these standards suggest that models should be developed "to predict and show relationships among variables between systems and their components in the natural and designed worlds" (NGSS Lead States 2013).

Supporting Individual Construction

Model-based inquiry takes as a fundamental premise that in most fields, including the sciences, experts are distinguished from novices not only by what they *know*, but also by the ways in which they *perceive* situations (Glaser and Chi 1988; Newell and Simon 1972; Simon and Chase 1973; Lesh and Doerr 2003a, b; Lesh et al. 2008). Thus, developing expertise in a domain resides at least as much in building powerful *interpretation systems* (also known as "models"), as it does in accumulating mastery over collections of facts and skills. This perspective helps to emphasize a view of learning that involves the construction, appropriation, and synthesis of new models, or the revision and expansion of the scope of locally applicable models to more general settings.

This research perspective is particularly tuned to producing descriptions of learners' ways of thinking and carefully documenting changes in their ideas during learning. Nevertheless, even among theoretical frameworks that take student thinking as a central focus, there is a range of potentially conflicting perspectives. For instance, historically, many technological designs for STEM education have focused on identifying and correcting student *misconceptions*. In these designs, researchers work to catalog common mistakes and misunderstandings about a given topic, to bring students into confrontation with their misconceptions, and then to offer an expert explanation or definition (Carey 1988; Clement 1982; Driver et al. 1994; McCloskey 1984; Posner et al. 1982; Tasar 2010; Trowbridge and McDermott 1980, 1981). While this strategy has proved enormously popular in educational design, it has for the most part proved ineffective, as studies have shown student misconceptions to be particularly "sticky" and resistant to change (see, e.g., McDermott 1983).

One reason for this stickiness is that "misconceptions" of this kind are in fact often *useful* in everyday situations, providing effective guidance for ordinary action. For example, while it is true that in a frictionless world an object in motion will remain in motion, real-world experience suggests objects in motion always slow down and eventually stop. For this reason, it is not surprising that students resist letting go of their "misconceptions" when "expert" explanations contradict thousands of observations and experiences! Furthermore, while domain experts often hold rich and detailed theories about a phenomenon, novices tend to apply ideas or explanations in the moment, depending on the "framing" of the specific situation (diSessa 1993, 1996; Hammer et al. 2005; Hammer 1996; Sherin 2006). So while an explanation given by a novice about a phenomenon may fit an identified "misconception," the same novice may offer an expert explanation of the same phenomenon in a different context (diSessa 1993).

In contrast to a misconceptions-centered approach, we have adopted a *knowledge-in-pieces* perspective for the design and analysis of our PNoM unit. This manifold model argues that cognition is emergent from many disparate, low-level knowledge elements that are highly sensitive to the learner's interactions with their environment (diSessa 1988, 1993; Minsky 1986). Which knowledge resources are activated depends greatly on the learner's perception of a given environment (diSessa and Sherin 1998). In some cases, relevant resources may be activated leading to an expert-like explanation or response. In others, situational cues may lead to less-relevant resources taking priority. In such instances, learners may offer explanations that would be categorized as "misconceptions," regardless of whether or not this explanation was productive (Smith et al. 1994). The goal of a knowledge-in-pieces approach, then, is to help to refine the way learners interpret situational cues so that they are more likely to activate productive knowledge resources. In this approach, intuition is valued and harnessed, rather than seen as a roadblock to be removed.

To facilitate the refinement of learners' knowledge systems and to enhance their perception of important situational cues, we adopt a *constructionist* approach in the design of model-based inquiry environments (Papert and Harel 1991; Papert 1980). Constructionist designs empower learners to take charge of their own learning through the construction of public artifacts that are personally meaningful. The act of construction, along with the process of sharing and critiquing these constructions, catalyzes the refinement and reorganization of internal knowledge structures (Caperton 2010; Kafai 1995; Noss and Hoyles 1996; Papert and Harel 1991; Sherin et al. 1993; White 1993; Wilensky and Reisman 2006; Wilensky 1996). Over the past 15 years, much work has been done exploring the potential of constructionist designs for model-based inquiry (Abrahamson and Wilensky 2004; Blikstein and Wilensky 2004, 2009; Levy et al. 2006; Sengupta and Wilensky 2005, 2009; Stieff and Wilensky 2003).

In settings where computational media are available, there is a natural resonance between such constructionist model-based inquiry and various forms of *computer programming*. Student-programmed computational models can act effectively as "thought-revealing artifacts" (Lesh et al. 2000), offering "both a model of student thinking about the situation and a model that represents how the students have integrated both interdisciplinary knowledge and the constraints and affordances of the problem context." (Martin et al. 2006, p. 389). In addition, the ability to easily "run" a computational model allows students themselves to quickly test and revise their constructions, further facilitating intrinsic motivation and learner self-assessment (Committee for the Workshops on Computational Thinking 2010, 2011; Lesh and Doerr 2000, 2003a, b, 2012; Martin et al. 2006).

Figure 1 offers a simplified, schematic account of the modeling process in such a learning environment. The learner's ideas develop through iterative cycles of construction and interpretation, throughout the process of creating the computational artifact or program. Within those cycles, running the program (the circular arrow) illuminates hidden consequences of the representation; reflecting on it (the thought bubble) stimulates changes to

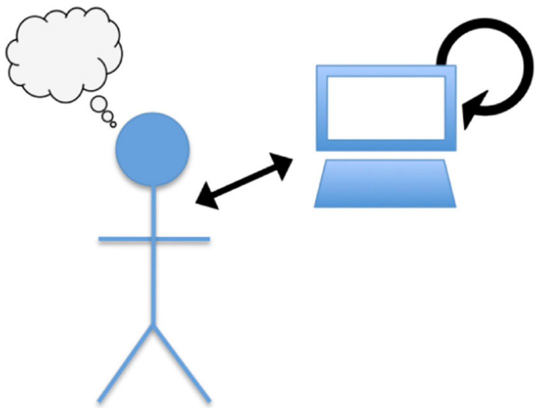


Fig. 1 Modeling in a computational medium

the learner's internal conceptions; and construction activity (the double-arrow) is motivated by disjunctions between the computational artifact and the learner's emerging intent or conception (which also includes the intended *referent*, not directly shown in the figure).

Where, then, are the *models* in this picture? For concreteness, we borrow a definition of "model" from Lesh and Doerr (2003a, b):

> Models are conceptual systems (consisting of elements, relations, operations, and rules governing interactions) that are expressed using external notation systems, and that are used to construct, describe, or explain the behaviors of other system(s)—perhaps so that the other system can be manipulated or predicted intelligently (p. 10).

In our case, the "external notation system" is the programmable computational medium. However, this definition does not suggest that the construction process is merely an externalization of a fixed internal conception. On the contrary, the work of externalization is itself a generative process—giving rise to changes in the learner's conceptions. As such, the nature of the external representation system used reflects back and has an effect on the nature of the conceptual systems of its author (diSessa 2000; Goody 1977; Wilensky and Papert 2006, 2010). We also see modeling as a multi-faceted, social process, encompassing the communication of emerging ideas, the negotiation of overlaps and conflicts among these ideas, and the integration of personal experiences with disciplinary forms of knowledge. Finally, at any given point in time, the learner's conceptions may not be complete, or the constructed external artifact may not adequate to articulate her conceptions. For all these reasons, we regard the learner's *model* as neither fully "in" the computational artifact nor fully "in" her head; rather, we view it as consisting of one or more representational artifacts the learner has produced, along with the collection of her intentions, explanations, and perspectives toward those artifacts.

At the same time that we are concerned directly with the emergence and refinement of this idiosyncratic sense-making process, we also must consider the models that have been created by the larger scientific community. The process of constructing a computational model involves an effort to capture the essence of a real-world system. For instance, in our PNoM unit, that system might be the phenomenon of diffusion or the behavior of a gas in a closed container. However, there are models from the disciplinary community of Chemistry that are also designed to describe these systems. The Kinetic Molecular Theory (KMT) would be a prime example, along with its quantitative entailments in the ideal gas laws. Thus, the modeling process of our learner takes place in relation to the modeling processes of a scientific community.

We apply the word "model" to both of these categories of conceptual systems—the idiosyncratic and the official. But it is often important to distinguish between these two types as well, if only to aid investigation into the relations that students are able to construct between them. In this article, we designate the former class of models, developed and articulated by learners themselves, with the phrase "*little-m*" *model*. For instance, in our PNoM unit, an example little-m model might be one student's expressions of contextual and case-specific understandings of particle interactions in matter, and her speculations about how observable phenomena might emerge from these interactions. When referring to the latter class of models, such as the KMT, we use the term "*Big-M*" *Model*. In contrast with little-m models, these conceptual structures are fundamental elements of the "paradigms" (Kuhn 1970) that characterize the ways of interpreting and conceptualizing the world which define entire scientific disciplines (Fig. 2).

Little-m models can be thought of as personal hypotheses or theories about how a system functions. While such models do not have the status of a Big-M model, to the learner these models are salient and useful for interpreting their immediate world. We view the interaction between models of these two types (Big-M and little-m) as a rich and important aspect of research into learning in model-based inquiry. In fact, a key element of the ESS hinges on the ability to place little-m and Big-M models in closer contact than indicated in Fig. 2. The ESS explicitly encodes a Big-M Model in the rules that govern the behavior of virtual objects added to its world. Thus, the Big-M Model actively governs the running of computational artifacts created in the ESS (Fig. 3).

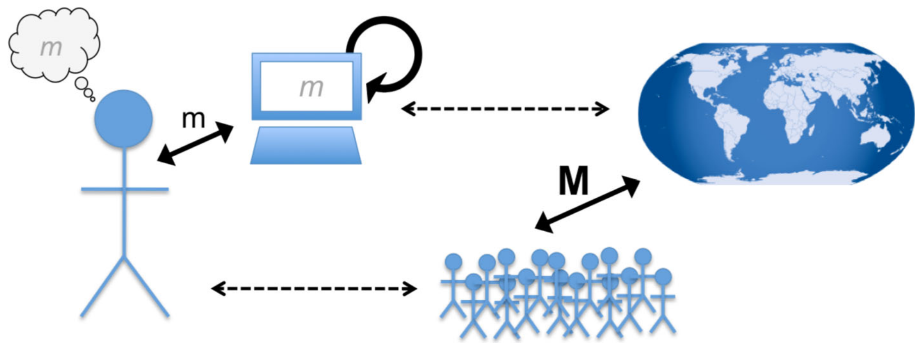


Fig. 2 An individual's (little-m) model in relation to a discipline's official (Big-M) model

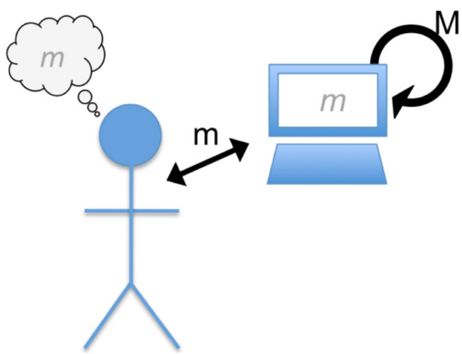


Fig. 3 Big-M and little-m models in an individual student's construction with an ESS

Designing for Social Interactions in Classroom Learning

As we have seen, computer programming not only fits the demands of model-based inquiry, but it also offers a powerful medium for supporting conceptual model development and refinement as suggested in the constructionist design paradigm. However, to the extent that this rich model of individual learning is pursued on its own, potential issues can emerge along both of the dimensions of "scale" that we have described above. For instance, the (numeric) scalability of learning designs that require teachers and students to become familiar with computer programming languages and their use may be questioned. This barrier can be particularly challenging to overcome in the context of classrooms where the primary subject area being studied is not computer science (Committee for the Workshops on Computational Thinking 2011; Guzdial 1994; Sherin et al. 1993). Although there is important ongoing research to address this very topic (Jona et al. 2014; Trouille et al. 2013; Sengupta et al. 2013; Wilensky 2014; Wilensky et al. 2014), this difficulty has also led constructionist learning scientists to search for ways to lower the threshold to authentic computational modeling, for example, by using visual rather than text-based programming tools (Sengupta et al. 2012; Wilkerson-Jerde and Wilensky 2010; Wilkerson-Jerde 2012).

At the same time, a programming-centered approach to model-based inquiry may also raise issues along the classroom-level dimension of scalability. Extensive, individualized construction projects may be difficult to manage in classroom, and it may not be straightforward to "unpack" student programs for their instructional value in STEM learning. Fellow students (or teachers, or researchers) who wish to understand the inner mechanisms of a text-based program must engage with it at the level of its code. This requires both a significant investment of time and a willingness to see the interpretive task of reading another's code as relevant to learning in the content area. Moreover, in a programming-centered learning environment, it may be difficult to engage students productively in sharing and analyzing the *interim* versions of each others' computational artifacts. The activity of debugging code can be an extremely rich learning activity, since unexpected behavior of a computational simulation often reflects "bugs" at the level of *thinking* as well as at the level of *coding* (Wilensky 2003); however, to engage in this activity, *socially* or collaboratively is a challenging proposition for classroom management.

Our theoretical framework attends to the social dimension of constructionist learning environments by drawing on the literature of *generative design* (Stroup et al. 2005, 2007; Davis 2010), which takes inspiration from the learning approaches of Wittrock et al. (Wittrock 1989, 1992; Osborne and Wittrock 1985). In these approaches, learning activities involve students in generating artifacts that reflect their emerging understanding. Stroup and colleagues build upon this work, identifying a gap that is closely related to our notion of classroom-level scaling. Specifically, they note that prior generative learning work has "underutilized the emergent space of behaviors and artifacts for classroom-based (group) learning and teaching" (Stroup et al. 2005, p. 191). In other words, students' independent work, *considered as a collective whole,* is seen as "space creating" in that it indicates the conceptual space of all potential responses. If this space can be visualized and discussed, it offers rich opportunities for collective discussion and reflection that prior designs did not exploit.

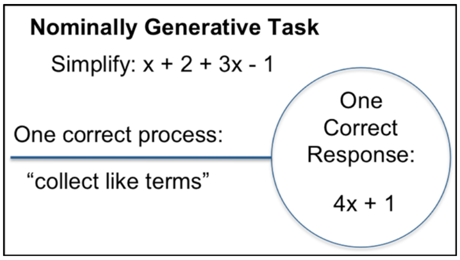


Fig. 4 Pathways-and-endpoints diagram for a nominally generative task

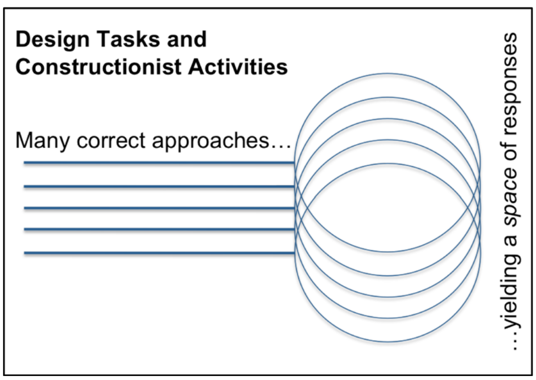


Fig. 5 Pathways-and-endpoints diagram for design tasks or constructionist activities

In working to fill this gap, Stroup and colleagues noted a resonance between their learning designs that engaged with classroom-level patterns in student work and the affordances of new classroom network technologies (Stroup et al. 2005; Stroup and Wilensky 2014). Such networks support a range of activity structures (Brady et al. 2013)—from PartSims (Wilensky and Stroup 1999b) (in which the entire class can enter a virtual world and interact as agents in a shared simulation) to the sharing of rich artifacts designed by individual students offline in parallel construction work. In the ModelSim project, we have provided support for this range of activity structures, by using the HubNet architecture (Wilensky and Stroup 1999a) for PartSims and by creating custom network technologies to support flexible sharing and review of computational artifacts.

To document and conceptualize the range of generative activity structures, Stroup et al. (2007) developed a "pathways-and-endpoints" task analysis framework that emphasizes the role assigned by an activity to the diversity of thinking in the classroom group. For instance, a traditional, single-right-answer task such as the algebra problem shown in Fig. 4 consists of a single endpoint (the correct answer) and a single pathway (or very few pathways) to

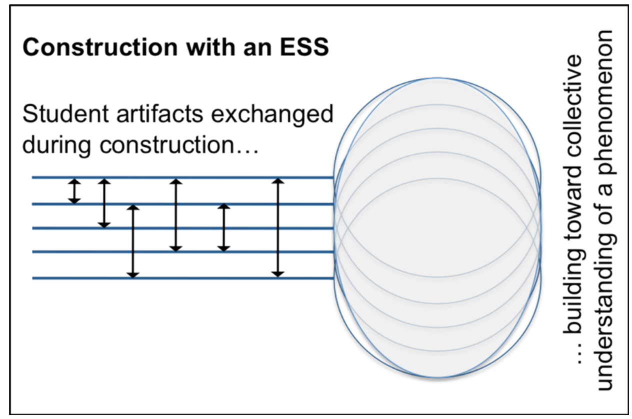


Fig. 6 Construction activity with an ESS

that endpoint (the correct or acceptable method(s)). Stroup and colleagues describe such tasks as "nominally generative."

In contrast, design tasks, including the tasks common to constructionist learning environments, may admit a wide range of both pathways and endpoints (Fig. 5). Here, the challenge is to make the diversity of different students' contributions and conceptions meaningful to one another so that they build to a collective understanding in which "the whole is greater than the sum of its parts."

With this pathways-and-endpoints framework in mind, we can articulate our strategies for increasing classroom-level scalability with the ESS. At the *endpoint* level, our activity design provides students with a shared experience—in the case of PNoM, the diffusion of an odor through the classroom—which students seek to illuminate or explain with their ESS constructions. Because all students work toward endpoints that are related through this shared representational goal, collections of student responses are necessarily relevant to one another as different perspectives on related facets of the phenomenon. At the *pathway* level, the ESS is designed to facilitate sharing of *in-process* artifacts. It does so both by enabling students to quickly make sense of each other's constructions (i.e., without grappling with them at the level of computer code) and by making it easy to share artifacts-in-process to foster peer commenting and feedback. In terms of the pathways-and-endpoints diagram, the ESS enables connectivity between pathways and the emergence of a collective endpoint (Fig. 6).

Finally, we can show the effect of attending to the social dimension in terms of our modeling diagram (Fig. 7). The individual modeling process of Fig. 3 is here augmented by a lateral, social process, driven by sharing and exploring the constructions of classmates. The act of sharing a construction itself stimulates further reflection, as Student 1 begins to move from m1 to m1′. Student 2, who is concurrently at work on her own construction (reflecting on

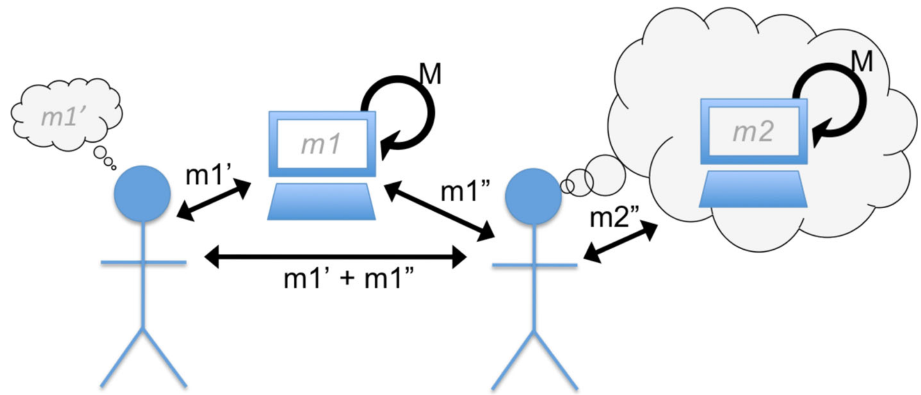


Fig. 7 Social dimension of modeling: students exchange in-process artifacts during construction

m2), downloads and interacts with Student 1's construction, forming a conception of it represented by m1″. Based on her (public) comments, Student 1 and 2 together make possible a new conception (m1′ + m1″). Moreover, as a result of the experience of reviewing her classmate's work, Student 2 begins to think differently about her own construction, giving rise to m2″, which she will incorporate into her future construction activity.

An ESS for PNoM

Earlier in the article, we defined an ESS as an agent-based computational modeling environment where construction primitives have been designed at the "entity" level to reflect an underlying Big-M Model; where learners can flexibly run their artifacts as they build them; and where learners can also easily save and share successive states of their constructions. In this section, we will go into more detail about the specifics of these components of the ESS definition and describe the particular ESS we produced for the first week of our PNoM unit, the Diffusion Sandbox.

Every computational modeling environment provides its users with a particular set of *primitives* from which larger models and systems are constructed. The primitives used in a modeling environment thus serve as a representational infrastructure for thinking through the components and mechanics of the models that are built in that environment. For this reason, in *designing* a modeling environment, the form of its primitives and the particular *level* at which these primitives exist are critical factors, which greatly determine the kinds of thinking and building that users can engage in when using the environment. To illuminate the design choice of defining the ESS primitives at the entity level, we give brief accounts of two other classes of modeling environment that use different levels of primitives, and we indicate the effects of those choices.

For example, in the NetLogo ABM environment, the properties and behaviors of agents are defined using a text-based programming language. Thus, when users of NetLogo "talk to" the agents in their models, they do so by issuing commands using this programming language. As such, the NetLogo building primitives exist at the programming language level, in the commands of the NetLogo language itself. By using primitives at this "low" level, NetLogo offers its users an extremely versatile toolkit for modeling systems across many domains, and it enables its users to define agent-level behaviors with arbitrary complexity. (Wilensky 2003; Tisue and Wilensky 2004). Moreover, learning environments created in NetLogo have the "glass box" feature that their mechanisms are always available in the "Code tab" for interested learners. This provides an always-available route to "high ceiling" inquiry to pursue questions about such simulation environments and even to modify or extend such environments.

Another class of modeling environments, including *Interactive Physics* (Roth 1995) or the PhET Interactive Simulations (Wieman et al. 2008), typically offers "higher" level primitives that focus construction activities at the level of human-scale objects. Here, the construction primitives might include virtual tools such as springs or motors for kinematics-based environments, or switches, ammeters, and jumper wires for electricity-based environments. Such environments are not particularly suited to modeling emergence and complexity, but because the human-scale object primitives are highly configurable, they can offer a powerful means for students to explore how these concrete real-world scientific apparatus work and to easily construct a wide variety of scenarios involving the possible configurations of these objects. Even when the programmatic mechanisms of these objects are black boxes to learners, they are similar enough to laboratory apparatus to invite further inquiry through exploring the functioning of these physical devices.

In contrast to environments that utilize programming language-level primitives and those that offer human-scale objects, an ESS provides users with primitives at an "entity" level. We define an "entity" as an agent-level object with fixed behaviors. Rather than allow users to program the rules that define agents' behaviors and interactions, in an ESS, these rules are predefined to encode the logic defined by the scientific (Big-M) model. Users add, arrange, and combine these entities to create working systems.

A consequence of designing primitives at this level is that the representational expressivity of an ESS is limited to situations that can be adequately understood through application of that Big-M Model. For instance, in our PNoM unit, the Big-M Model behind our Diffusion Modeling Sandbox ESS is the KMT. A student could *attempt* to utilize this ESS to model a phenomenon for which the KMT is not a useful model—say, snow falling in a forest. But such a student will find that his "snowflakes" travel forward at a uniform speed until they collide elastically with each other or with his "trees." While this student may be frustrated by the inability of this environment to produce a satisfactory model of snowfall, he may actually learn something powerful about the KMT as a result of this "failure." And herein lies one of the advantages of an ESS. Because every entity added to its virtual world relentlessly follows the rules of the Big-M Model, the consequences of these rules tend to become salient to users who create artifacts within the ESS. Moreover, because the rules are not open to editing, a construction in an ESS will *always run,* and it will always behave faithfully to the Big-M Model. It may not produce the aggregate behaviors that its author intends, but it will always produce outcomes that are the logical consequence of the Big-M Model, gradually nudging the learner's intuitions into alignment.

The nature of an ESS's primitives also distinguishes it from "macro-level" modeling environments. The ESS's entities exist at the agent level of the created system, and so the ESS is well suited to produce complex, emergent phenomena. Moreover, because these entities are not directly linked to human-scale objects, different students can make different choices about *what* human-scale objects they will build in their ESS constructions and *how* they will build these objects out of the entity primitives. In spite of the fact that there are a limited number of entity types, all of them having fixed, predefined behaviors, these features make it common for students to be surprised by each others' representational choices and intrigued by their ingenuity. Here again, the nature of the ESS primitives correlates with the learning affordances of the environment.

The design and level of the primitives of an ESS also support two distinct and powerful modes of interaction with the environment and enable the learner to move freely between these modes. One mode is immersive, "embodied" modeling (Wilensky and Reisman 2006), in which the learner explores what it is like to exist within the ESS environment and builds intuitions about the interactions of entities. In this mode, representations in the model are executed *continuously*: the simulation logic can be run *while* the user adds and manipulates entities in the ESS. Because in an ESS learners are building at the "entity" level, rather than at the "language" level or at a higher, aggregate level, it is particularly important that they attend closely to the objects that compose a particular structure or phenomenon. A major task of modeling in an ESS is orchestrating situations for agent-level entities to interact, and so these interactions and their consequences become salient for learners and are often the focus of iterative construction work. In an ESS environment, users do not discover the rules that govern a complex system by creating and coding them; they do so by putting entities that encode these rules into contact with one another, observing the results and making predictions about how the results might be different under alternative conditions.

In contrast to the immersive, continuous mode of interaction and execution, a second mode is external, comparative, and *discrete*. Here, the learner wishes to observe the emergent behavior of a system that they have constructed, to ensure alignment with their target phenomenon. In this mode, it is important to pause execution, establish the arrangement and conditions of the dynamic entities, and then cause the environment to run for a period of time to test and revise conjectures. Moreover, because the two activities of intuition-building (continuous execution) and conjecture testing (discrete execution) are fluid, both modes must be present in an ESS and learners should be able to quickly and easily transition between the two.

Finally, the ESS's facilities for saving and sharing the states of constructions also follow from the nature of the primitives and the two interaction modes described above. To support the discrete mode of interaction, it is vital that learners are able to establish "initial" conditions in the virtual world, "run" this construction, and then return to the prior state to make modifications for further explorations. Flexible saving of states supports discrete executability in enabling this iterative process.

But state saving (and publishing) also enables an entirely new dimension of exploration, at the social level. Because there are only a few entity types in an ESS, students find each others' constructions comprehensible on visual inspection. While surprise occurs as a result of different choices of *how* to use a given entity to construct a given aggregate object or phenomenon, by the very definition of the ESS, the entities themselves and their behaviors are part of a shared lexicon. On the other hand, because these constructions exhibit complex behaviors, students can gain significant insight into the nature of the entities and of the underlying Big-M Model by interacting with their peers' constructions. Moreover, because students

Fig. 8 Data visualizations from three moments in the whole-class odor-diffusion experience

know their peers are building models of the same shared phenomenon as they are, they have an incentive to study and respond to each others' perspectives on the phenomenon, by downloading and interacting with their constructions. In this peer review process, both the continuous and discrete modes of execution gain another level of utility, in supporting student's understanding and assimilation of ideas from each others' little-m models in process.

In the remainder of this section, we advance the description of the ESS construct by introducing the particular ESS we developed for the PNoM unit of the ModelSim project. This ESS, called the Diffusion Modeling Sandbox, was used extensively in the first week of the unit. On the first day of the unit, students engage as a whole-class group with an experience of odor-diffusion in their physical classroom. Two containers of perfume are introduced into the classroom space—one heated and the other at room temperature. The students are told that they will act as smell sensors, recording the changing intensity of odor that they detect overtime. Knowing that the experiment can only be run once (it is hard, if not impossible, to "reset" one's nose or clear the room completely), the group discusses how best to arrange themselves in the room, and how to record their sense data. After running the experiment, the class explores a visualization of the intensity-level data that they have recorded and discusses what they see (Fig. 8). These discussions typically involve identifying patterns in the spatio-temporal spread of the scent, observing fluctuations or exceptions in those patterns, thinking about causal factors such as temperature, and discussing various ideas about mechanisms that could explain the spread of the odor.

The ESS is introduced immediately after this inconclusive but idea-generating whole-class discussion. Student pairs are asked to explore the ESS environment and use it to develop an expressive, executable model that explains one or more aspects of the shared experience of the diffusion phenomenon. Figure 9 shows an image of the initial state of this ESS, with its chooser list of available primitive "entities" expanded. As you can see, the environment provides a fairly open canvas for student construction, offering only a collection of green particles for a start (in fact, even this initial feature can be removed or altered by the learner, by adjusting the *initial-#-particles* slider and pressing the *setup* button). The set of construction tools simply suggests that student work could involve creating "walls" of various kinds (fixed and removable, of different colors), particles of different colors, and sensors that detect each of the different particle types. There are also tools for changing particles' color, for speeding up or slowing down specified particles, and for removing entities that have been created in the sandbox. To add or modify entities in the environment, learners simply select in the chooser the appropriate action and "paint" the objects or changes directly into the sandbox space.

Execution of constructions in the Diffusion Sandbox is controlled by the *pause-particles?* switch. When particles are unpaused, they immediately and continuously execute their rules of motion. Regardless of the state of the *pause-particles?* switch, the learner can add, modify, or remove entities from the construction. This enables both continuous and discrete interactions with the environment, as described above. Saving, sharing, and loading of states are controlled by other buttons in the interface, which work in conjunction with a web-based Gallery that we constructed for the project, where learners can view, download, and comment on each other's posted states.

In spite of the unstructured and open interface of the Diffusion Modeling Sandbox, the system elements that the learner can construct all behave according to rules that reflect the underlying Big-M Model of KMT. In particular, in the sandbox, particles collide elastically with each other and with walls of all kinds, and walls provide surfaces against which collisions are also elastic, and where reflections obey a rule equating angles of incidence and reflection. Because these micro-, agent-level behaviors and features are analogous to rules that learners observe in macro-level interactions (e.g., billiard balls and their

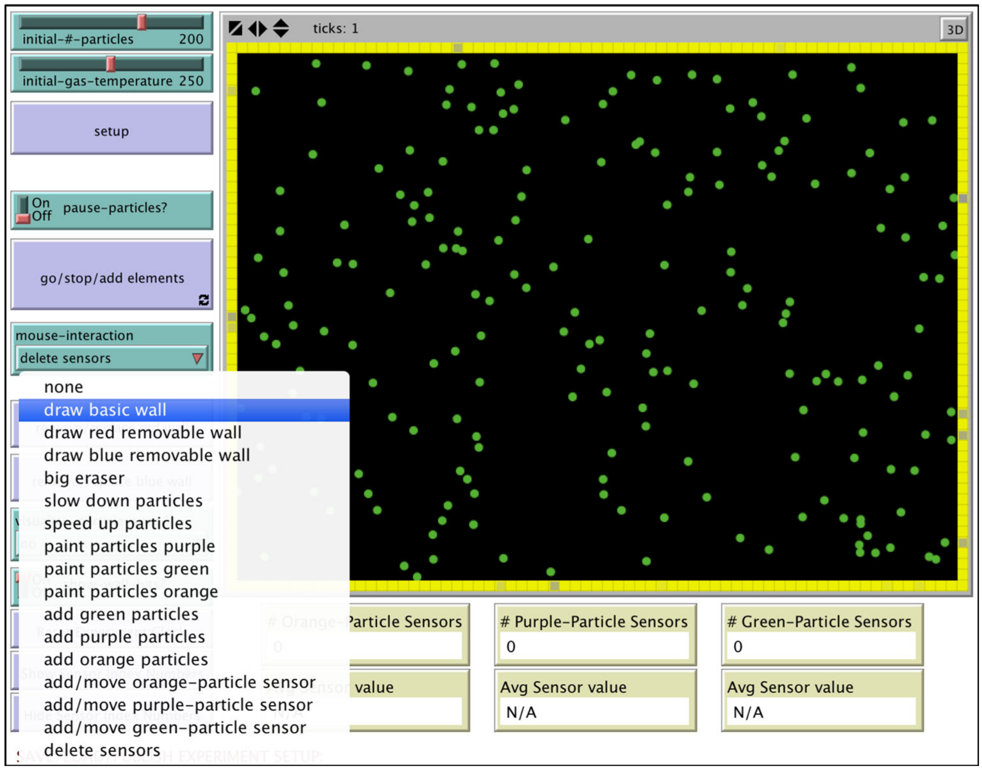


Fig. 9 The initial state of the Diffusion Sandbox environment, and the construction primitives available to the builder

behavior in bouncing off walls and each other), learners are able to draw upon intuitions from their everyday lives in working with these objects in the sandbox. At the same time, the essence of the KMT is that a system of many agents obeying these simple and intuitive rules can give rise to emergent phenomena (Wilensky 2001)—including not only diffusion but also other aggregate-level properties such as pressure and temperature. Students' constructions within the sandbox environment require them to engage deeply with the various aspects of this underlying Big-M Model to achieve their personal little-m goals. Model development in this ESS environment is thus an entry-level process of discovering the expressive potential of the agent-level components and their ability to account for and produce these aggregate phenomena. This construction process is interactive, as the Sandbox can provide continuous feedback to the learner by executing the agent-level behaviors of the components that the learner constructs and arranges. It is iterative, as the learner can save states of the artifact as it develops, to test the effect of changes in a systematic way. And it is social, in that these states and the modeling process as a whole occur in a classroom community of learners who are producing, sharing, and reviewing each others' computational artifacts, which offer different perspectives on a phenomenon that the group has experienced together.

In the remainder of the article, we will describe and analyze data from implementations of the PNoM unit with the Diffusion Modeling Sandbox, showing how the design components of the ESS gave rise to key features of student activity and interaction conducive to effective, scalable, model-based inquiry.

Methods

The data presented below are drawn from implementations conducted by two teachers that taught in two different high schools, each serving a diverse population in the metro area of a large Midwestern city. Before implementing the PNoM unit, these teachers were trained on the technology and activity structures during a 2-day summer workshop. Researchers were present during the implementation to offer both technical and instructional support. The implementations described here occurred in two honors and one regular chemistry class with students in their freshmen to junior year.

A large corpus of data was collected, including both whole-classroom and "roving" video data; answers to online questions before, after, and during the unit implementation; as well as computational artifacts and models created by students during the unit. While the non-model-building activities of the unit certainly impacted the larger experience, in this paper we focus specifically on the ways learners engaged the tools available in the ESSs, the artifacts created within the ESS environments, and their conversations around these artifacts.

Features of the ESS in Action

In this section, we describe three features of the experience of constructing within an ESS, illustrating our discussion with examples from students' use of the PNoM unit's Diffusion Modeling Sandbox. In our implementations, we have found that the ESS provides students with a *referentially underdetermined* space for exploration; that it supports flexible modeling through its dual modes of continuous and discrete executability; and that it affords students a variety of opportunities for validating their qualitative understandings of emergent systems as these understandings develop.

Opening a Referentially Underdetermined Space for Exploration

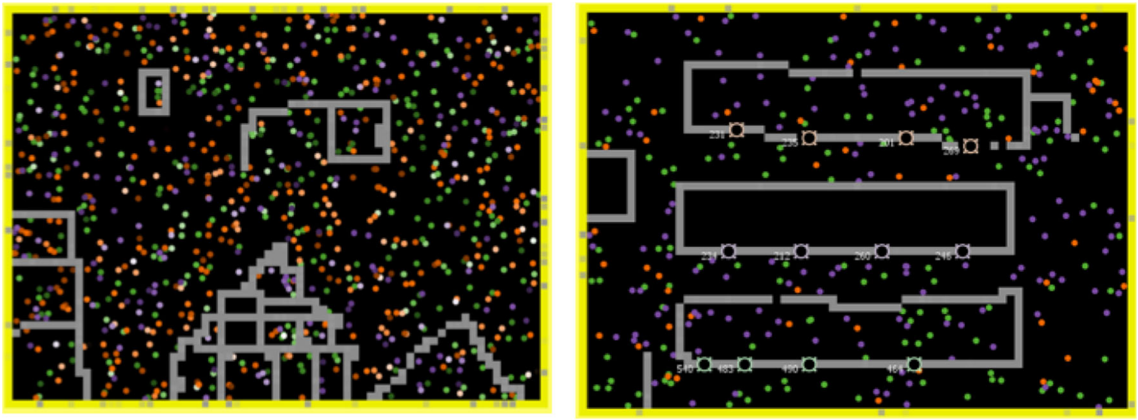
The construction primitives of an ESS are intentionally designed to be *referentially underdetermined*, in the sense that they do not map singly or obviously to apparatus in the observable world. For one thing, they exist at a lower, more fundamental level than things such as desks, flasks, or perfume. Thus, macro-level, human-scale objects must be built out of aggregations of these entities. But there is also nothing in the ESS that predetermines the mappings between the entity types and macro-level objects. The names, such as "walls" and "particles," do provide suggestive guidance, but learners are free to explore possible mappings.

Designing an ESS to be referentially underdetermined engages the learner in two important and linked layers of interpretation that drive the modeling process. In the first layer, learners work to apply meaning to the individual *objects* and entities; and in the second, learners try to make sense of the *behaviors* of these objects, in terms of their referents. So while learners work to identify what the "green particles" could represent, they must also make sense of how and why these particles move in the way they do. Why do they move in straight lines? Why do they bounce and reflect off of walls? As learners begin to apply a referent to objects in the model, they bring with this referent a certain set of possible behaviors. Likewise, as learners interpret behaviors, a certain set of objects become possible referents and others are excluded. These two sense-making activities occur simultaneously and are complementary, in that each act supports and constrains the other; and they are central in helping the learner to connect their personal little-m model with the Big-M Model targeted by the designer.

To encourage students to explore the space of possible references in the Diffusion Sandbox Model, we explicitly incorporated free exploration time into our PNoM implementations. During the first day of explorations, we found that students often produced constructions that were like drawings, rather than immediately attempting to model the target phenomenon of diffusion. While these drawings varied dramatically, most converged on images and systems that included particulate phenomena such as "snow" (Fig. 10a). Though initially it might have seemed that learners were neglecting "more important" work in creating such drawings, we came to recognize the importance of this "messing about" (Hawkins 1974) phase. Indeed, during this time, learners began to develop their intuitions about how the objects in the sandbox world could be manipulated. Their ideas were reflected in the particular phenomena they chose and the ways they used "animation" in their drawings. As intuitions about virtual objects and their behaviors grew, this in turn increased students' readiness to employ the virtual objects as referents. By the end of the first day, while students may still have been "drawing," these drawings began to take on the features and characteristics of objects relevant to the phenomena under study, such as including the layout and features of the classroom (Fig. 10b).

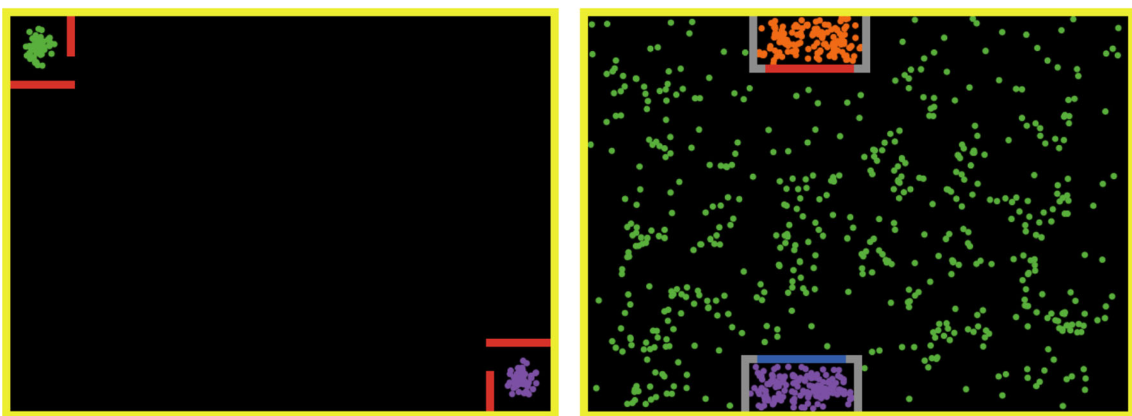
In an ESS, this "seeing-as" step—seeing a rectangular arrangement of wall patches as the perfume container in a shared diffusion experiment, for example—is initially left up to the learner and is a critical part of the construction process. As learners begin to see the ESS as a medium in which to simulate scientific phenomena, they begin to tune their constructions toward specific questions (or factors) about how these phenomena unfold. Often students first define one object or construction as a central figure (such as drawing a box out of wall entities and deciding that it represents a container of perfume) and then gradually apply a set of references to objects in relation to that first mapping. While some students may exhibit a literalism in their constructions, such as reproducing the classroom along with the rows of lab-tables (Fig. 10b), others abandon strict literalism in an attempt to create conditions where a particular mechanism or outcome will show itself most clearly. Some of these "schematic" designs suggest attempts to isolate variables and test-specific experimental conditions, such as exploring the differences between diffusion with and without air present (Fig. 11).

While learners individually define the mappings between ESS entities and what these entities come to represent, over the course of an implementation *whole classes* begin to develop a shared understanding and expectations of what ESS entities are and can be. For example, classroom groups may coalesce around particular strategies for representing macro-level objects such as "containers" and "desks," or they may begin to develop classroom *norms* such as "we usually use green particles to represent air." While each of these particular examples has been observed in our PNoM implementations, due to the



(a) (b)

Fig. 10 Early explorations included drawings of interesting pictures or representation of particulate phenomena such as "snow" (**a**). Later constructions more literally modeled the room where the diffusion phenomena took place (**b**)



(a) (b)

Fig. 11 Examples of a schematic and experimental approach—exploring how particles would diffuse with (**b**) and without (**a**) air

ambiguity of the objects and entities that can be added in the Diffusion Sandbox Model, we have found a great diversity *between* different classes.

This emergence of a common language and expectations around referents shared generally by each class as a whole is an effect of the social dimension of the ESS, which promotes communication about ideas that the group feels are important: in this case the normative mappings of the environment's "underdetermined" entity primitives. To facilitate multiple rounds of exploration and design, as well as to encourage students to share and collaborate during model development, we developed the means for students to "go public" with their discoveries by publishing the state of their ESS worlds to a shared "Gallery" at any time they chose. Once a world-state is posted, anyone in the class can view a snapshot of the world, comment on it, or load the full executable state into their own sandbox environment for testing, exploration, or refinement. Because students are encouraged to post in-progress model designs frequently to the gallery and are explicitly directed to explore the models created by their classmates, students must necessarily *read* and *interpret* the models of others.

Making sense of others' models facilitates a shared understanding of entity referents. As students begin to comment on or discuss others' models, they freely apply their own mapping to entities and objects in models created by others. For example, after observing and running another student's construction, one student posted a comment on the Gallery to the model author stating, "I think you should have made the heated molecules and cold molecules separate colors. Also, I like how you incorporated air molecules." While the model authors did not explicitly state what each object or entity in the model was meant to represent, the commenting student immediately mapped "heated" and "cold molecules" to particular groupings of circles present in the model and also assumed the circles distributed throughout the model must be "air molecules." For students that had not considered how the presence of air particles might impact the modeled phenomena, this mapping suggested by another student may seem reasonable and lead them to incorporate this feature in their own models (doing so on the assumption that this interpretation of green-circles-as-air-particles is correct), further facilitating the spread of this particular mapping.

By designing primitives to be representationally underdetermined, we have also created an opportunity for students to be surprised by how others might use an entity in an innovative way. While comments similar to the one cited above certainly could occur in other modeling environments, the entity level at which ESS building primitives exist greatly determines the form and purpose of these exchanges. For example, when engaged in modeling at the language level, one may expect conversations and comments around others' constructions to focus on algorithmic or behavioral rule-level choices, such as the decision to use fixed directions for a particle's motion rather than incorporating an element of randomness. Alternatively, when building in an environment such as *Interactive Physics*, the macro-nature of the building primitives may instead give rise to discussions that center around the relative correctness or the effects of connecting various macro-level objects with particular configuration settings, similar to how one might discuss the appropriateness of a particular equipment assembly in a laboratory setting. In contrast, the use of referentially underdetermined entities allows students to attend to each others' reference mappings and the effects of these choices: both to recognize mappings and to be surprised by them. This shifts the focus of discussions from being about either the nuances of programming or the appropriate use of scientific apparatus to center on both the novelty of agent-to-aggregate constructions and the value of particular entity mappings.

Supporting Flexible Model Construction with Continuous and Discrete Executability

To create experiences that both facilitate intuition-building and encourage learners to move toward hypothesis generation, ESSs allow learners to build constructions that are both *continuously* and *discretely executable*. Executable representations in computational media are designs that exhibit dynamic behavior in response to logic encoded in their construction. They thus "run" semi-independently of their human authors, which allows those authors to reflect on their behavior and/or to interact with them as they run. For instance, an executable representation in the Diffusion Sandbox Model might be a *closed container* that only releases particles when triggered to open, or a *vent* that heats and pushes particles as they approach. While each is constructed using entity-level primitives, their meaning and function exist at a higher "object level."

While almost all computer programming and modeling environments include executability of some sort as a core feature, the nature and the degree of this executability are significant points in the design of such environments. Moreover, different types and levels of executability may be appropriate for different modes of exploration or different phases in model construction. In a similar way, recent trends in computing that offer a more fluid interaction between the programmer and the programmed environment, sometimes known under the heading of "live coding" (Burg et al. 2013), suggest a new, more dynamic relationship between programmer and program that may augment and complement the more traditional "express-test-revise" (Martin et al. 2006) debug cycle common to most existing programming environments. In describing the executable representations available within an ESS, we highlight the advantages of both a *continuous* and a *discrete* mode of execution.

The continuous mode of executability is akin to the "live coding" style described above. And though live coding is a fairly novel affordance in computer programming environments, continuously executable representations have been studied in other areas of the learning sciences for over two decades. For instance, in the research literature on mathematics learning, the value of such representations was explored in the 1990s as access to computational power sufficient to support real-time executability ushered in what Balacheff and Kaput described as "a new experiential mathematical realism" (1997, p. 470). Here, the term "realism" refers to the *reification* of mathematical constructs as virtual objects that become manipulable (e.g., through interaction with "hotspots") while obeying the structural constraints and relations that define them. The 1990s witnessed the growth of dynamic mathematics software tools such as Kaput's own SimCalc MathWorlds (Kaput and Roschelle 1996) environment for the study of the math of change and variation; the Geometric Supposer (Schwartz and Yerushalmy 1987), Cabri Geometry (Laborde 1990), and the Geometer's Sketchpad (Jackiw 1991) for dynamic geometry; and Fathom (Finzer et al. 2002) and later TinkerPlots (Konold and Miller 2005) for the dynamic study of statistics, probability, and data modeling.

An ESS enables this type of "experiential realism" in modeling by supporting a construction mode in which entities move and interact *while* the learner is in the process of building. As such, an ESS has some key features in common with dynamic mathematics software and especially with modern dynamic geometry environments. In particular, both support the construction of systems of virtual objects and relations whose dynamic interactions obey the constraints and specifications of an underlying Big-M Model. The fluid exchanges between learner and environment enabled by such a design permit constructive modeling processes characterized by what Moreno-Armella et al. call *cognitive partnership* (Moreno-Armella and Sriraman 2005) and *co-action* (Moreno-Armella and Hegedus 2009). According to this perspective, the direct engagement with objects that embody the rules and structures of a discipline offers learners a continuous, dialogic relation with those knowledge structures. In this way, live coding in computer science, hotspot dragging in dynamic geometry software, and systems construction in an ESS all offer exciting possibilities for co-action in computationally mediated environments. These possibilities hinge on the creation of a virtual space that affords the user continuous interactions with dynamic objects in a world where disciplinary structures define the phenomenology of those interactions.

In our work with ESS environments, we have observed the power of such spaces for facilitating student sense-making, perhaps most notably in the early stages of exploratory use. In these early interactions, learners attempt to gain an intuitive understanding of the way the virtual world works. Papert (1980) described such activities as "getting to know" a virtual world and argued for the potential of even text-input-driven *microworlds* for building this type of qualitative understanding of the world's structures (e.g., p. 137). While we believe it is certainly possible to engage in this intuition-building outside of a continuous execution environment, we argue that the fluid interactions facilitated by such settings are particularly *resonant* with this intuition-building objective and that therefore the ESS environment's support of this mode is a strong affordance. Finally, while we have highlighted the value of continuous executability here in the context of individual construction, it has a significant value for the social dimension as well. The ability to *inhabit* and *explore* a fellow student's construction offers a compelling incentive to engage in peer review, and the vividness of the experience encourages students to appropriate key elements of the constructions that they explore in this way.

An ESS also supports another key mode of executability, the *discrete* mode, which is conducive to investigations of another kind. In general, we observe students' interactions with ESS environments to move from what we call "present-tense" focused work (e.g., "when I do X, Y happens"), which is supported well with continuous executability, to "subjunctive" or "future-tense" focused work (e.g., "under condition X, Y will happen"), which requires a more punctuated, discrete form of execution. Investigating the behavior of a system at this level requires the ability to control conditions, repeat experiments, and observe results. To support this shift, our ESS makes it possible not only to pause the dynamic running of the environment, but also to *save the state* of the representational world and to reload this state at a later time. This enables learners to *design and develop experiments* that carry their intuitive understandings to the next level and produce more robust and shareable results. While "present-tense" thinking explores the nature of agent-level interactions in an ESS, this new "future-tense" style can involve observing and reasoning about the aggregated effects of these interactions overtime. Moreover, this discrete mode of execution is particularly powerful along the social dimension of modeling. As learners seek to share their findings with others (or even to align their interpretations of virtual phenomena within a single working group), they are led to formalize their observations and move toward a future-tense style, supported in this effort by the ability to save, load, and share world states publicly in the Gallery.

Both continuous and discrete modes of executability are critical to the investigations and constructions facilitated by our PNoM ESS. Continuous executability supports the rapid growth of students' intuitions both about the Big-M Model of the KMT and about the little-m models reflected in their classmates' constructions. And as understanding grows, discrete executability enables students to articulate and share predictions about the aggregate-level behavior of their virtual constructions, substantiating them through the design of informal but replicable experiments.

Offering the Means for Validating Qualitative Understandings

A third key feature of ESS environments is that they offer learners a variety of opportunities to seek and receive validation for the little-m models expressed in their constructions, as they develop. An important concern with open constructionist learning designs is the question of whether, in the language of this article, students' little-m models eventually *converge* on the Big-M Model that represents normative disciplinary understandings of key phenomena. There are many approaches to ensuring such convergence, but the mechanism we emphasize in our ESS design is that of external validation. That is, providing occasions for subjecting the qualitative impressions that learners have about the emergent behaviors of their computational artifacts to increasingly rigorous standards of evidence. In this section, we describe two means by which students can receive such validation: (1) adding mechanisms to their constructions that produce quantitative measurements of phenomena, and (2) negotiating meaning and resolving conflicting ideas in a social space, among constructions identified as describing the same or overlapping phenomena.

The first means of validation, adding quantitative measurements, is conducive to the kind of "future-tense" thinking described at the end of the previous section. An ESS can provide students the ability to *instrument* their virtual environments in various ways; in the PNoM ESS, this includes adding "particle detectors" of different kinds or measuring changes in selected virtual objects or regions. By introducing and arranging these sources of quantitative data, students can substantiate qualitative impressions that they have about emergent properties of their constructions. Moreover, the ESS provides the ability to *plot* some of these data overtime. In combination with the ability to save and reload states, these quantification features support learners in quantifying observations, substantiating assertions, and validating predictions about their virtual experiments. For example, after adding virtual sensors to her ESS, one student commented:

> Increase in temperature helps speed up molecule diffusion. The peppermint at the front of the room (which was heated) was detected faster and a lot more strongly than the peppermint at the back of the room (lower temperature), which was not detected at all. The particles that were sped up were detected more by the sensors (purple in our case), and the slower particles not as much. Particles that were sped up bounced around more and passed over the sensors more often. Particles that had a higher temp bounced around more and with more force thus giving them a better chance of being detected by the sensors.

Here, the student's comments blend claims about real-world diffusion phenomena with evidence and argumentation about their virtual construction. The conflation of statements about physical and virtual worlds suggests that she has begun to see the mechanisms of her virtual construction as offering a (little-m) model of the physical world experience. The sensor readings over multiple runs of her construction have provided her with a warrant for her claims about the relation between temperature and diffusion rates.

A second means of external validation in the ESS arises in the social space. At a basic level, students are aware that their construction efforts are related to one another, since they are all attempting to create models of an aspect of the shared diffusion experience. Moreover, because students publicly post their works-in-progress, they are able to identify relations and overlaps between their construction project and those of their classmates. In terms of the pathways-and-endpoints diagram, the shared experience promotes connectivity between the endpoints of the students' construction efforts, while the ease of sharing supports students in identifying affiliated work that their peers are doing. Both forms of connectivity between students' work support an increased sense of need to articulate their findings clearly and precisely, as their fellow students constitute an engaged and authentic audience for these findings.

The social dimension of the PNoM ESS modeling environment in fact has a prominent role from the very beginning of the unit, when students act as "smell sensors" to produce a collective representation of the diffusion phenomenon that the classroom can continually refer to and reflect on throughout the course of the unit (Fig. 8). In the visualization produced by this activity, each "dot" represents the data collected by a single student. However, the *significance* of these data is only recognized in the emergent patterns that appear *among* the dots, both in space and overtime. Thus, the "agent-level" testimony of the individual student must be placed in a social context for it to have meaning. Moreover, students' engagement with the ESS environment is explicitly framed as an effort to investigate, reproduce, or explain some features of this shared physical experience in their virtual sandboxes. In this way, students' independent investigations are implicitly linked in a communal effort to understand diffusion in general and the particular phenomena that the class has experienced as a group.

To build upon this social foundation of inquiry, students are encouraged to post works-in-progress and completed models in the ESS Gallery (as described in the section on the referentially underdetermined feature of the ESS) as well as to run and comment on the models posted by others. We argue that the existence of a ready-to-hand means for "going public" provides an authentic audience for the students' work and applies an appropriate pressure to articulate their intuitive understanding of the model in more formal ways—specifically, as observations and communicable findings. Furthermore, due to the entity-level primitives used in ESS models, students can easily read and interpret classmates' models without needing to make sense of complex computer code. By removing these roadblocks to understanding, students are free to attend to the common or distinctive elements among their classmates' constructions. In our implementations, we found evidence of learners comparing and contrasting various designs of virtual experiments in written comments posted to the gallery and during whole-class discussions, and we observed spontaneous identification of shared lines of inquiry emerging among multiple student groups working in parallel. Such evidence suggests that this social dimension was an important aspect of learners continued refinement of their little-m models.

When commenting on others' models in the gallery, we found that students often offered endorsements or critiques of explorations or experimental designs by directly comparing the posted model to mechanisms and designs found

Fig. 12 Posting to the diffusion gallery

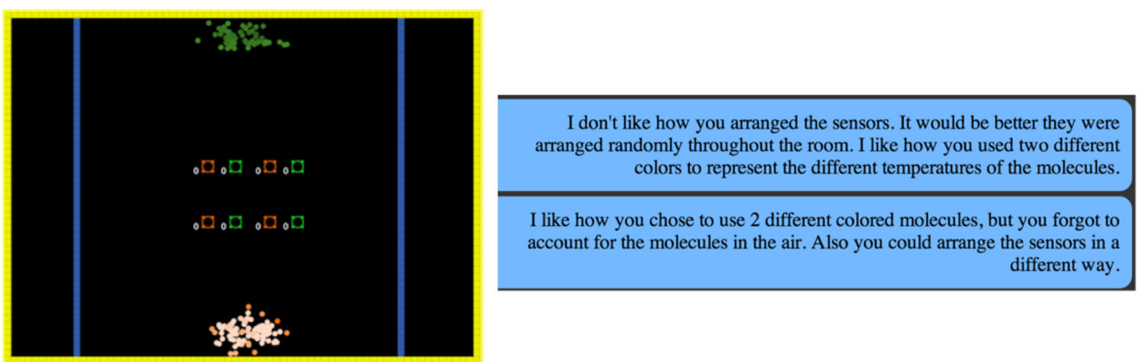


Fig. 13 Model produced by the author of the first comment on Fig. 12, and comments on his model from two other classmates

in the author's own work. For example, the first of the comments seen in Fig. 12 reads, "I am kind of confused as to why your diffusion starting points are in the corners and why the sensors are not together. However, good job implementing the air molecules." Behind this comment stands a reflective and discursive history. The author of this comment was from a group that did *not* include air molecules in their representation and arranged their sensors in a grid at the center of the virtual room. Comments on *their* posting, from still other groups, had critiqued their orderly, centralized sensor arrangement and had also noted: "…you forgot to account for the molecules in the air" (Fig. 13).

Thus, this group's comment in Fig. 12 acknowledges the value of an alternative perspective on the question of air molecules, but maintains a critical stance on the question of sensor positions. Because this commentary occurs in a public space, however, the conversation continues. Subsequent commenters on the model in Fig. 12 pick up these themes, expressing different perspectives on sensor placement, while also reinforcing the value of including air molecules in the representation:

> Your experiment was very different than ours. I liked how you placed the sensors all in different places to show where the particles reached most….

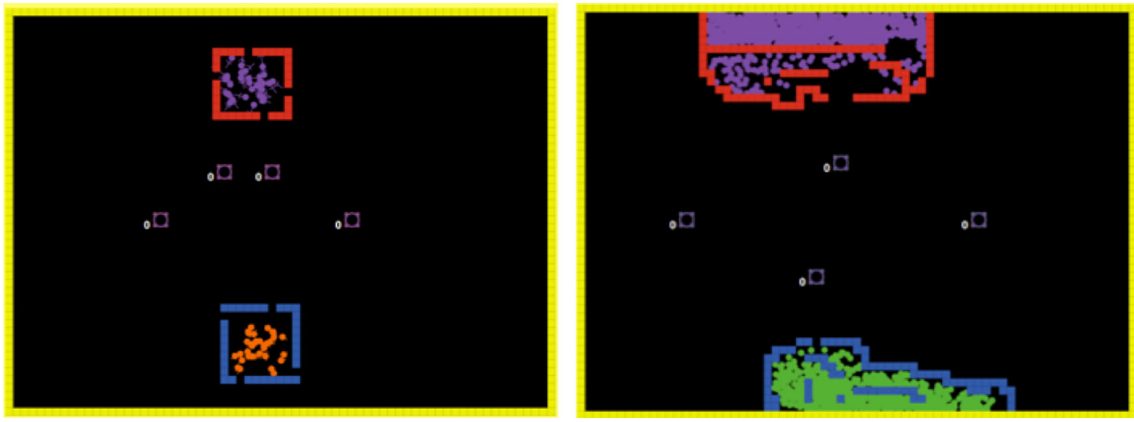


Fig. 14 Introducing structures within the perfume containers to slow the escape of particles

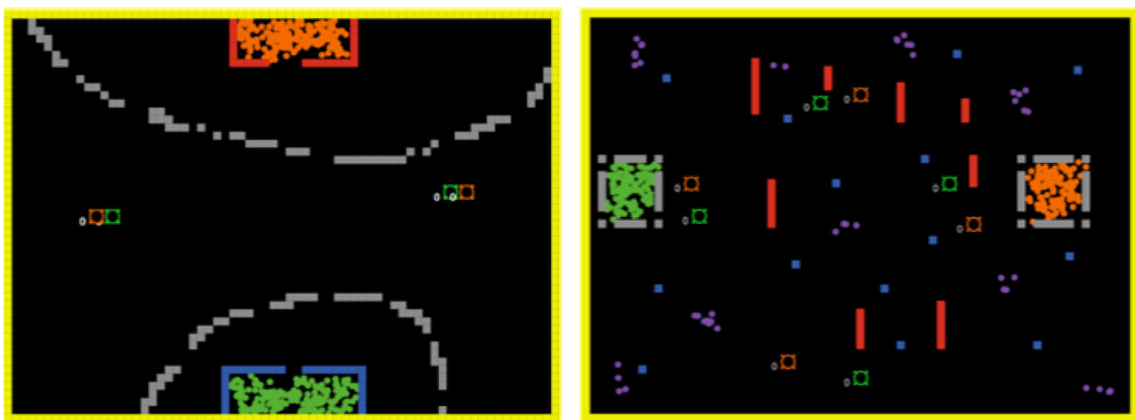


Fig. 15 Constructing barriers to slow the spread of particles

> I like how the areas containing the purple and orange particles are the same size. I also like how your model contains what looks to be air molecules.

Later, in the teacher-facilitated, whole-class discussion of the models, issues associated with including or omitting air molecules were raised as well, with different student groups explaining their decisions and the impact they felt these decisions had. Interestingly, in this discussion, the question of the validity of a model without air was *not* settled simply through the assertion that "the world is not like that," though verisimilitude or fidelity to the real world was certainly important to some students. Rather, student groups that had designed a vacuum argued that while realism was certainly sacrificed in this sense, working with a vacuum allowed them to more closely investigate other aspects of particle behavior. That is, these constructions maintained their status as models of particle diffusion, while departing from being models of the particular phenomenon and conditions of perfume diffusion as experienced in the opening classroom experiment.

The existence of an ongoing channel for communication between the groups and the ability to survey classmates' strategies-in-process seems also to have encouraged the spontaneous formation of groups with shared research interests and lines of inquiry. For instance, a number of students in this class became interested in or concerned about the rapidity of diffusion in the virtual world and wanted to introduce mechanisms that would produce the slow and gradual diffusion that they had observed in the physical world during the classroom experiment. To do this, some of these groups introduced structural features in their virtual containers to slow the exit of particles, as in the work shown in Fig. 14. Other students, pursing the same area of interest, created barriers in the open area to affect the rate at which particles moved through the space, as in Fig. 15.

This spontaneously formed interest-based subgroup of the class provides a microcosm that illustrates how groups' independent work supported the emergence of collective understandings. Because different groups explored different facets of the diffusion phenomenon, or explored the same facet from different perspectives, the class as a whole covered more ground than any individual group. This enabled whole-class discussions to be an opportunity for external validation as the class worked to fit individual groups' findings together, identifying both complementarity and conflict, and creating the basis for a shared model of the diffusion phenomenon. While it is still in principle possible for learners at the end of the unit to have an interpretation of the dynamics of the ESS that is at odds with the KMT, this is much less likely when their findings are integrated with those of their classmates in constructing a shared descriptive model.

Furthermore, teachers can make strategic use of whole-class discussions to capitalize on group resources that develop over these iterative cycles of model construction and peer commenting. In the ModelSim project, we have recognized the fundamental role of the teacher in facilitating consensus-building discussions to draw upon the distributed insights that the classroom group has encountered. In this way, teachers can support the class in creating shared expressions of disciplinary core ideas, as facets of the underlying "Big-M" model of the KMT that are directly connected to the group's collective construction work. Our professional development has focused specifically on alerting teachers to these opportunities and on exploring high-leverage teacher moves to facilitate such discussions.

Conclusion

In this article, we have introduced the design construct of an ESS. We have described its role in ongoing research into the *scalability* of innovative constructionist approaches to ABM and model-based inquiry. Here, we defined scalability along two dimensions: first, as the ability for a model of implementation to succeed in its learning goals in classroom settings where the multifarious support of researchers and designers, characteristic of laboratory studies, is not present. Such scalability allows an innovation to be applied in a greater *number* of classrooms and under more diverse and variable conditions. Our second dimension of scalability referred to the ability of a classroom implementation to make use of the social resources of the classroom group in materially enhancing the learning experience. Such scalability allows an approach to move beyond an individual experience and register its effects on the shared understandings of the classroom group—on their collective ways of talking about and practicing science. We have argued that these two dimensions can in fact be linked through learning designs that support students in *going public* in meaningful ways with their in-process findings to an authentic peer audience. Building on these ideas of scalability, we described our vision of the models and the modeling processes that can be studied in classroom settings using an ESS.

We also discussed three key elements of the classroom experience of the ESS that we designed for our PNoM unit. First, we showed how constructions of *representationally underdetermined* entities enabled creative exploration of a range of possible phenomena that could be produced with particulate dynamics governed by the KMT. At the individual level, we showed how the construction process dialectically balanced explorations of entity dynamics with explorations of possible referents or mappings of the entities in the ESS to aspects of phenomena of interest (in this case, phenomena associated with the diffusion of odors). And at the social level, we described the emergence of shared interpretations and shared conventions among the students in particular classroom groups. Second, we showed how the discrete and continuous forms of executability of the ESS facilitated both an immersive experience of the KMT "world" and the ability to run increasingly controlled and repeatable experiments on virtual constructions. We indicated how these features supported understanding and argumentation at both the individual and social levels. And third, we discussed the role of external validation in the modeling process. This occurred through increasing use of instrumentation and measurement in students' constructions; through discursive reference to interpretations of the shared diffusion experience; and through recognition and resolution of perceived conflicts or incompatibilities between the constructions of different student groups. Finally, we suggested how the teacher plays an essential role in capitalizing on the learning opportunities and insights that arise as the classroom group engages with the ESS.

Each of the features of the ESS that we have discussed in this article was carried forward beyond the diffusion experiment and into the second week of the PNoM unit. In the course of that instructional sequence, the initial ESS was augmented with new components (such as movable platforms that responded to particle impacts and release valves to remove particles from constructed systems); increased tools for instrumentation and measurement (including tools to measure particle concentrations, the variable or constant volume of enclosed regions, and so forth); additional means of programmatic control (including the ability to heat or cool specified regions and to inject particles into specific regions at specified points in time); and further opportunities to share constructions and results with the classroom group. With these supports, students extended their particulate models of diffusion to build understandings of the behavior of gases under varying temperature, pressure, and volume. Finally, in a culminating engineering task, they used the KMT principles and mechanisms studied over the course of the unit to construct hybrid virtual-physical machines designed to achieve particular tasks or outcomes (e.g., to drive a saw, hammer a nail, propel a hydraulic catapult, or power a circular paddle wheel). In this construction, they used the augmented ESS to construct virtual pistons; this virtual motion was translated in real time to a coupled physical device that then powered Lego-based machines that the students built.

Future work in the direction of this article will involve more extensive analyses of data we are collecting of students' work with ESS environments, across a wider range of classroom settings. Such analyses will enable us to identify connections between the learning opportunities and resources generated through construction work with ESSs and particular patterns in student inquiry behaviors, teacher moves, and other implementation features. They will also enable us to link work within the ESS with quantitative measures of student learning.

We argue that an approach to learning design centered on ESS environments can place the practice of *modeling* at the center of a generative social process of inquiry, while also engaging other key scientific practices identified in the NGSS Framework. Indeed, in the course of the implementations of the PNoM unit, we have observed student activity with and within ESS environments that engage with each of the eight core practices of science and engineering identified by the Framework. It is our hope that our experiences in designing and implementing ESS environments as expressed in this article might offer insights into ways in which technology can be used to enhance the processes of learning through models and modeling in experiences that support *scalability* in both of the senses we have defined.

Acknowledgments The research reported here is based upon work supported by the National Science Foundation under Grant #DRL-1020101. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the NSF.

References

Abrahamson D, Wilensky U (2004) ProbLab: a computer-supported unit in probability and statistics. In: Proceedings of the 28th annual meeting of the international group for the psychology of mathematics education. Bergen, Norway

Balacheff N, Kaput J (1997) computer-based learning environments in mathematics. In: Bishop AJ, Clements K, Keitel C, Kilpatrick J, Laborde C (eds) International handbook of mathematics education. Springer, Netherlands, pp 511–564. Retrieved from http://link.springer.com/chapter/10.1007/978-94-009-1465-0_15

Blikstein P, Wilensky U (2004) MaterialSim: an agent-based simulation toolkit for learning materials science. In: International conference on engineering education. Gainsville, Florida

Blikstein P, Wilensky U (2009) An atom is known by the company it keeps: a constructionist learning environment for materials science using agent-based modeling. Int J Comput Math Learn 14:81–119

Brady C, White T, Davis S, Hegedus S (2013) SimCalc and the networked classroom. In: Hegedus S, Roschelle J (eds) The SimCalc vision and contributions: democratizing access to important mathematics. Springer, New York, NY, pp 99–121

Buckley BC, Gobert JD, Kindfield A, Horwitz P, Tinker R, Gerlits B, Willett J (2004) Model-based teaching and learning with BioLogica: what do they learn? how do they learn? how do we know? J Sci Educ Technol 13:23–41

Burg B, Kuhn A, Parnin C (2013) 1st international workshop on live programming (LIVE 2013). In: Proceedings of the 2013 international conference on software engineering, pp 1529–1530 Retrieved from http://dl.acm.org/citation.cfm?id=2487068

Caperton IH (2010) Toward a theory of game-media literacy: playing and building as reading and writing. Int J Gaming Comput Mediat Simul 2(1):1–16

Carey S (1988) Reorganization of knowledge in the course of acquisition. In: Ontogeny, phylogeny, and historical development. Ablex, Norwood, NJ, pp 1–27

Clement J (1982) Students' preconceptions in introductory mechanics. Am J Phys 50:66–70

Committee for the Workshops on Computational Thinking (2010) Report of a workshop on the scope and nature of computational thinking. National Research Council, Washington, DC

Committee for the Workshops on Computational Thinking (2011) Report of a workshop on the pedagogical aspects of computational thinking. National Research Council, Washington, DC

Davis SM (2010) Generative activities: making sense of 1098 functions. In: Lesh R, Galbraith PL, Haines CR, Hurford A (eds) Modeling students' mathematical modeling competencies. Springer, New York, NY, pp 189–198

diSessa AA (1988) Knowledge in pieces. In: Constructivism in the computer age. Lawrence Erlbaum Associates, Inc., Hillsdale, NJ, pp. 49–70

diSessa AA (1993) Toward an epistemology of physics. Cogn Instr 10:105–225

diSessa AA (1996) What do "just plain folk" know about physics? In: The handbook of education and human development: new models of learning, teaching, and schooling. Blackwell, Oxford, pp 709–730

diSessa AA (2000) Changing minds: computers, learning and literacy. The MIT Press, Cambridge, MA

diSessa AA, Sherin B (1998) What changes in conceptual change? Int J Sci Educ 20:1155–1191

Driver R, Squires A, Rushworth P, Wood-Robinson V (1994) Making sense of secondary science: research into children's ideas. Routledge, New York, NY

Finzer W, Erickson T, Binker J (2002) Fathom dynamic statistics software. Key Curriculum Press Technologies, Emeryville, CA

Glaser R, Chi MTH (1988) Overview. In: The nature of expertise. Lawrence Erlbaum Associates, Inc., Hillsdale, NJ, pp xv–xxvii

Gobert J, Horwitz P, Tinker B, Buckley B, Wilensky U, Levy ST, Dede C (2003) Modeling across the curriculum: scaling up modeling using technology. In: Paper presented at the twenty-fifth annual meeting of the cognitive science society, Boston, MA

Goody J (1977) The domestication of the savage mind. Cambridge University Press, Cambridge, MA

Guzdial M (1994) Software-realized scaffolding to facilitate programming for science learning. Interact Learn Environ 4(1):001–044

Hammer D (1996) Misconceptions or p-prims: how may alternative perspectives of cognitive structure influence instructional perceptions and intentions? J Learn Sci 5:97–127

Hammer D, Elby A, Scherr RE, Redish EF (2005) Resources, framing, and transfer. In: Transfer of learning from a modern multidisciplinary perspective. Information Age Publishing, Greenwich, CT, pp 89–120

Hawkins D (1974) The informed vision: essays on learning and human nature. Algora Publishing, New York, NY

Jackiw N (1991) The geometer's sketchpad [Software]. Key Curriculum Press Technologies, Emeryville, CA

Jona K, Wilensky U, Trouille L, Horn MS, Orton K, Weintrop D, Beheshti E (2014) Embedding computational thinking in science, technology, engineering, and math (CT-STEM). In: Paper presented at the future directions in computer science education summit meeting, Orlando, FL

Kafai YB (1995) Minds in play: computer game design as a context for children's learning. Lawrence Erlbaum Associates, Hillsdale, NJ

Kafai YB (2006) Constructionism. In: The Cambridge handbook of the learning sciences. Cambridge University Press, New York

Kaput J, Roschelle J (1996) SimCalc MathWorlds. University of Massachusetts, Dartmouth, MA

Konold C, Miller CD (2005) TinkerPlots: dynamic data exploration. Key Curriculum Press, Emeryville, CA

Kuhn TS (1970) The structure of scientific revolutions. University of Chicago Press, Chicago

Laborde JM (1990) CABRI Geometry [Software]. Brooks-Cole Publishing Co, New York, NY

Lehrer R, Schauble L (2006) Cultivating model-based reasoning in science education. In: Cambridge handbook learning sciences, pp 371–388

Lesh R, Doerr H (2000) Symbolizing, communicating, and mathematizing: key components of models and modeling. In: Symbolizing and communicating in mathematics classrooms. Lawrence Erlbaum Associates, Hillsdale, NJ, pp 361–384

Lesh R, Doerr H (2003) Foundations of a models and modeling perspective on mathematics teaching, learning, and problem solving. In: Beyond constructivism: a models & modeling perspective on mathematics teaching, learning, and problem solving. Lawrence Erlbaum Associates, Hillsdale, NJ

Lesh R, Doerr H (eds) (2003b) Beyond constructivism: a models & modeling perspective on mathematics teaching, learning, and problems solving. Lawrence Erlbaum Associates, Hillsdale, NJ

Lesh R, Doerr H (2012) Alternatives to trajectories and pathways to describe development in modeling and problem solving. In: Blum W, Ferri RB, Maaß K (eds) Mathematikunterricht im Kontext von Realität, Kultur und Lehrerprofessionalität. Springer Fachmedien Wiesbaden, Germany, pp 138–147

Lesh R, Hoover M, Hole B, Kelly A, Post T (2000) Principles for developing thought-revealing activities for students and teachers. In: Kelly A, Lesh R (eds) Handbook of research design in mathematics and science education. Lawrence Erlbaum Associates, Mahwah, NJ, pp 591–646

Lesh R, Hamilton E, Kaput J (eds) (2008) Models & modeling as foundations for the future in mathematics education. Lawrence Erlbaum Associates, Hillsdale, NJ

Levy ST, Wilensky U (2009a) Crossing levels and representations: the Connected Chemistry (CC1) curriculum. J Sci Educ Technol 18:224–242

Levy ST, Wilensky U (2009b) Students' learning with the Connected Chemistry (CC1) curriculum: navigating the complexities of the particulate world. J Sci Educ Technol 18:243–254

Levy ST, Novak M, Wilensky U (2006) Students' foraging through the complexities of the particulate world in the Connected Chemistry (MAC) curriculum. In: Annual meeting of the American educational research association, San Francisco. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.170.3899&rep=rep1&type=pdf

Martin F, Hjalmarson M, Wankat P (2006) When the model is a program. In: Lesh R, Hamilton E, Kaput J (eds) Foundations for the future in mathematics education. Lawrence Erlbaum Associates, Hillsdale, NJ

McCloskey M (1984) Naive theories of motion. In: Mental models. Lawrence Erlbaum, Hillsdale, NJ

McDermott LC (1983) Critical review of research in the domain of mechanics. In: First international workshop research on physics education. Paris, pp 139–182

Minsky M (1986) The society of mind. Simon & Schuster, New York

Moreno-Armella L, Hegedus SJ (2009) Co-action with digital technologies. ZDM Math Educ 41(4):505–519. doi:10.1007/s11858-009-0200-x

Moreno-Armella L, Sriraman B (2005) The articulation of symbol and mediation in mathematics education. ZDM Math Educ 37(6):476–486

Newell A, Simon HA (1972) Human problem solving, vol 14. Prentice-Hall, Englewood Cliffs, NJ

NGSS Lead States (2013) Next generation science standards: for states, by states. The National Academies Press, Washington, DC

Noss R, Hoyles C (1996) Windows on mathematical meanings: learning cultures and computers. Kluwer, Dordrecht

Osborne R, Wittrock M (1985) The generative learning model and its implications for science education. Stud Sci Educ 12:59–87

Papert S (1980) Mindstorms. Basic Books, New York

Papert S, Harel I (1991) Situating constructionism. In: Constructionism. Ablex Publishing, New York

Passmore CM, Svoboda J (2012) Exploring opportunities for argumentation in modelling classrooms. Int J Sci Educ 34(10):1535–1554

Posner GJ, Strike KA, Hewson PW, Gertzog WA (1982) Accommodation of a scientific conception: toward a theory of conceptual change. Sci Educ 66(2):211–227. doi:10.1002/sce.3730660207

Roth WM (1995) Affordances of computers in teacher–student interactions: the case of interactive physics™. J Res Sci Teach 32(4):329–347

Schwartz JL, Yerushalmy M (1987) The geometric supposer: an intellectual prosthesis for making conjectures. Coll Math J 18(1):58–65

Schwarz CV, Reiser BJ, Davis EA, Kenyon L, Achér A, Fortus D, Krajcik J (2009) Developing a learning progression for scientific modeling: making scientific modeling accessible and meaningful for learners. J Res Sci Teach 46(6):632–654

Sengupta P, Wilensky U (2005) NIELS: an emergent multi-agent based modeling environment for learning physics. In: Proceedings of the 4th international joint conference on autonomous agents and multi-agent systems (AAMAS). Utrecht, Netherlands

Sengupta P, Wilensky U (2009) Learning electricity with NIELS: thinking with electrons and thinking in levels. Int J Comput Math Learn 14:21–50

Sengupta P, Farris AV, Wright M (2012) From agents to continuous change via aesthetics: learning mechanics with visual agent-based computational modeling. Technol Knowl Learn 17(1–2):23–42

Sengupta P, Kinnebrew JS, Basu S, Biswas G, Clark D (2013) Integrating computational thinking with K-12 science education using agent-based computation: a theoretical framework. Educ Inf Technol 18:351–380

Sherin B (2006) Common sense clarified: the role of intuitive knowledge in physics problem solving. J Res Sci Teach 43:535–555

Sherin B, diSessa AA, Hammer D (1993) Dynaturtle revisited: learning physics through the collaborative design of a computer model. Interact Learn Environ 3:91–118

Simon HA, Chase WG (1973) Skill in chess: experiments with chess-playing tasks and computer simulation of skilled performance

throw light on some human perceptual and memory processes. Am Sci 61(4):394–403
Smith JP, diSessa AA, Roschelle J (1994) Misconceptions reconceived: a constructivist analysis of knowledge in transition. J Learn Sci 3(2):115–163. doi:10.1207/s15327809jls0302_1
Stewart J, Cartier JL, Passmore CM (2005) Developing understanding through model-based inquiry. In: How students learn: science in the classroom. The National Academies Press, Washington, DC, pp 515–565
Stieff M, Wilensky U (2003) Connected chemistry—incorporating interactive simulations into the chemistry classroom. J Sci Educ Technol 12:285–302
Stroup W, Wilensky U (2014) On the embedded complementarity of agent-based and aggregate reasoning in students' developing understanding of dynamic systems. Technol Knowl Learn 19(1–2):1–34
Stroup W, Ares N, Hurford A (2005) A dialectic analysis of generativity: issues of network-supported design in mathematics and science. Math Think Learn 7(3):181–206
Stroup W, Ares N, Hurford A, Lesh RA (2007) Diversity by design: the what, why and how of generativity in next-generation classroom networks. In: Lesh RA, Kaput JJ (eds) Foundations of the future: twenty-first century models and modeling. Lawrence Erlbaum, New York, NY
Tasar MH (2010) What part of the concept of acceleration is difficult to understand: the mathematics, the physics, or both? ZDM Math Educ 42:469–482
Tisue S, Wilensky U (2004) NetLogo: design and implementation of a multi-agent modeling environment. In: Proceedings of the agent 2004 conference on social dynamics: interaction, reflexivity and emergence, Chicago, IL
Trouille L, Beheshti E, Horn M, Jona K, Kalogera V, Weintrop D, Wilensky U (2013) Bringing computational thinking into the high school science and math classroom. In: American astronomical society meeting abstracts, vol 221
Trowbridge DE, McDermott LC (1980) Investigation of student understanding of the concept of velocity in one dimension. Am J Phys 48:1020–1028
Trowbridge DE, McDermott LC (1981) Investigation of student understanding of the concept of acceleration in one dimension. Am J Phys 49:242–253
White BY (1993) ThinkerTools: causal models, conceptual change, and science education. Cogn Instr 10(1):1–100. doi:10.2307/3233779
Wieman CE, Adams WK, Perkins KK (2008) PhET: simulations that enhance learning. Science 322(5902):682–683
Wilensky U (1996) Making sense of probability through paradox and programming: a case study in a connected mathematics framework. In: Constructionism in practice: designing, thinking, and learning in a digital world. Lawrence Erlbaum, Mahwah, NJ
Wilensky U (1999a) NetLogo [computer software] version. Center for connected learning and computer-based modeling, Northwestern University, Evanston, IL. http://ccl.northwestern.edu/netlogo
Wilensky U (1999b) GasLab: an extensible modeling toolkit for exploring micro-and-macro- views of gases. In: Roberts N, Feurzeig W, Hunter B (eds) Computer modeling and simulation in science education. Springer, Berlin, pp 151–178
Wilensky U (2001) Modeling nature's emergent patterns with multi-agent languages. In: Proceedings of EuroLogo 2001. Linz, Austria
Wilensky U (2003) Statistical mechanics for secondary school: the GasLab modeling toolkit [special issue]. Int J Comput Math Learn 8(1): 1–4
Wilensky U (2014) Computational thinking through modeling and simulation. In: Whitepaper presented at the summit on future directions in computer education. Orlando, FL. http://www.stanford.edu/~coopers/2013Summit/WilenskyUriNorthwesternREV.pdf
Wilensky U, Papert S (2006) Restructurations: reformulations of knowledge disciplines through new representational forms. In: Annual meeting of the American educational research association, San Francisco
Wilensky U, Papert S (2010) Restructurations: reformulations of knowledge disciplines through new representational forms. In: Proceedings of constructionism 2010 Paris, France
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. Cogn Instr 24:171–209
Wilensky U, Stroup W (1999a) HubNet. Center for connected learning and computer-based modeling, Northwestern University, Evanston, IL. http://ccl.northwestern.edu/netlogo
Wilensky U, Stroup W (1999) Learning through participatory simulations: Network-based design for systems learning in classrooms. In: Proceedings of the 1999 conference on computer support for collaborative learning, CSCL '99 Palo Alto, CA
Wilensky U, Levy S, Novak M (2004) Connected chemistry curriculum. Retrieved from http://ccl.northwestern.edu/curriculum/ConnectedChemistry/
Wilensky U, Brady C, Horn M (2014) Fostering computational literacy in science classrooms. Commun ACM 57(8):17–21
Wilkerson-Jerde MH (2012) The DeltaTick project: learning quantitative change in complex systems with expressive technologies. Retrieved from http://dl.acm.org/citation.cfm?id=2522404
Wilkerson-Jerde MH, Wilensky U (2010) Restructuring change, interpreting changes: the deltatick modeling and analysis toolkit. In: Proceedings of constructionism 2010. Paris, France
Windschitl M, Thompson J, Braaten M (2008) Beyond the scientific method: model-based inquiry as a new paradigm of preference for school science investigations. Sci Educ 92(5):941–967
Wittrock M (1989) Generative processes of comprehension. Educ Psychol 24(4):345–376
Wittrock M (1992) Generative learning processes of the brain. Educ Psychol 27(4):531–541