URI WILENSKY

STATISTICAL MECHANICS FOR SECONDARY SCHOOL: THE GASLAB MULTI-AGENT MODELING TOOLKIT *

ABSTRACT. In the Connected Learning projects, we are studying students' learning of content through exploring and constructing computer-based models of that content. This paper presents a case study of a high school physics teacher's design and exploration of a computer-based model of gas molecules in a box. We follow up the case study with shorter vignettes of students' exploration and elaboration of the Gas-in-a-Box model. The cases lead us to analyze and discuss the role of model-based inquiry in science and mathematics education as well as to draw some general conclusions with respect to the design of modeling languages and the design of pedagogies and activities appropriate for model-based inquiry in classroom settings.

1. INTRODUCTION

Several high school students are huddled around a personal computer, eyes intent on the screen. They watch as little blue, green and red shapes speed across the screen, bouncing off a containing "box" and colliding with each other and changing speeds. One shouts, "that slow molecule just sped up real fast when it hit the other one. Why does it do that?" The others nod in agreement with his puzzlement and they all start to suggest computer "experiments" that could help them answer the question. "Collide a bunch of them all from the same locations and see if it's always the same". "Do the slow ones always speed up after a collision, or do they sometimes slow down even more?"

These high school students at a run-of-the-mill urban high school in Massachusetts are engaged in heated discussion of statistical thermal physics. This content, considered amongst the hardest topics for graduate students in physics, is made accessible to these students through the medium of a computer-based modeling environment, NetLogo (Wilensky, 1999), in which the students can explore, experiment and analyze the interactions of large numbers of simulated molecules.

* A version of this paper was reprinted in Wilensky, U. (1999). GasLab – an Extensible Modeling Toolkit for Exploring Micro- and Macro-Views of Gases. In N. Roberts, W. Feurzeig and B. Hunter (Eds), *Computer Modeling and Simulation in Science Education*. Berlin: Springer Verlag. The present paper is, however, significantly revised and updated due to a five year delay in publication.

International Journal of Computers for Mathematical Learning **8:** 1–41, 2003.

In this paper, I will present several vignettes of a secondary teacher and students constructing and exploring NetLogo models of statistical mechanical phenomena. These vignettes illustrate the powerful learning these so-called average students achieved through their model-based inquiry. The students' explorations led to the development of a set of models, dubbed GasLab, that collectively form a toolkit enabling exploration and further construction of gas molecule models. The argument will be made that multi-agent modeling toolkits such as GasLab enable a much larger and younger segment of society to engage with the powerful ideas of statistical thermal physics and thereby obtain much deeper explanations of traditional secondary physics/chemistry content as well.

We begin in the next two sections with an introduction to dynamic systems modeling. The subsequent two sections present the case studies and vignettes that are the heart of the paper. We then move on to discuss general pedagogical and design issues educators face when we introduce model-based inquiry into learning environments and classroom settings.

2. DYNAMIC SYSTEMS MODELING

Computer-based modeling tools have largely grown out of the need to describe, analyze, and display the behavior of dynamic systems. During recent decades, there has been a recognition of the importance of understanding the behavior of dynamic systems – how systems of many interacting elements change and evolve over time and how large-scale patterns can arise from local interactions of these elements. New research projects on chaos, self-organization, adaptive systems, nonlinear dynamics, and artificial life are all part of this growing interest in systems dynamics. The interest has spread from the scientific community to popular culture, with the publication of many general-interest books about research into dynamic systems (e.g., Barabasi, 2002; Bonabeau et al., 2001; Buchanan, 2002; Gell-Mann, 1994; Holland 1998; Johnson, 2001; Kauffman, 1995; Kelly, 1994; Roetzheim, 1994; Waldrop, 1992).

Research into dynamic systems touches on some of the deepest issues in science and philosophy – order vs. chaos, randomness vs. determinacy, analysis vs. synthesis. The study of dynamic systems is not just a new research tool or new area of study for scientists. The study of dynamic systems stands as a *new form of literacy*, a new way of viewing, describing, and symbolizing phenomena in the world. The language of the present mathematics and science curriculum employs *static* representations. Yet, our world is, of course, constantly changing. This disjunct between the world of dynamic experience and the world of static school representations stands as one source of student alienation from traditional curricula (Bertalanffy, 1975; Stroup, 2002; Wilensky, 1997b). The gap between students' experience of the world around them and mathematical representations of that experience has grown even larger as students increasingly play in virtual worlds and games (Turkle, 1995; Bruckman, 1994; Kafai, 1998) which make use of dynamic representations. Such dynamic representations, theoretical perspectives and computer-based tools arising out of the study of dynamic systems can describe and display the *changing* phenomena of science and the everyday world. Moreover, these tools can do so in ways that are accessible to large numbers of students. There is therefore now an opportunity to enable many more students to engage in genuine science and mathematics inquiry, to explore and characterize how systems unfold and change over time, to uncover the *dynamics* at work that create seemingly static patterns in nature and society and, through such inquiry, to bring this new form of literacy, a literacy in the dynamics and patterns of change, to the great majority of students and citizens.

3. DYNAMIC SYSTEMS MODELING IN THE CONNECTED LEARNING PROJECTS

At the Center for Connected Learning and Computer-based Modeling, over the past fifteen years,[1] we have conducted a large number of "Connected Learning" projects in which student learn a variety of content domains through computer-based modeling. Among these are the Connected Probability project (Wilensky, 1995a, 1995b, 1996, 1997b; Wilensky and Resnick, 1999; Wilensky and Stroup, 2000; Abrahamson and Wilensky, 2003), the ConnectedScience project (Wilensky and Reisman, 1998, in press), the Connected Chemistry project (Stieff and Wilensky, 2002, 2003), the Connected Evolution project (Centola et al., 2000; Centola and Wilensky, 2000) and a number of others. The goal of these projects is to study learners (primarily high school students) engaged in model-based inquiry of scientific phenomena. As part of these projects, learners are provided with access to a variety of modeling tools that they can use in pursuit of their investigations. They make particular use of the computer-based modeling languages StarLogoT[2] (Wilensky, 1995a, 1997b) and NetLogo[3] (Wilensky, 1999) to conduct their investigations.[4]

StarLogoT and NetLogo are instances of a new class of computer-based modeling languages known as "multi-agent" languages (aka object-based parallel modeling languages). Such languages enable users to visualize, explore and construct computer-based models of a wide variety of worldly phenomena. Users can create these models by giving rules of behavior to large numbers of so-called “agents,” graphical objects that can move about the computer screen and interact. StarLogoT and NetLogo are multi-agent languages specifically designed to enable users without modeling experience to explore and construct models. Both these languages are extensions of the Logo language in which a user controls a graphical turtle by issuing commands, such as “forward,” “back,” “left,” and “right” and they are descendents of the StarLogo[5] language. In StarLogoT and NetLogo, the user can control thousands of graphical turtles. Each turtle is a self-contained “agent” with internal local state. Besides the turtles, these languages automatically include a second set of agents, “patches.” A grid of patches undergirds the environments’ graphics window. Each patch is a square or cell that is computationally active. Patches have local state and can act on the “world” much like turtles. Essentially, a patch is just a stationary turtle. For any particular model, there can be arbitrarily many turtles (from 0 to 32000 is the range found in the models we have used), but there are, typically, a fixed number of patches (e.g., something like 10,000 laid out in a 100×100 grid).[6]

Over the years of the Connected Learning projects, several different versions of multi-agent Logo languages have been developed. The first implementation ran on a connection machine, a parallel supercomputer. Parallel emulation implementations soon ensued for UNIX workstations and personal computers. Each of these implementations preserved the essential ingredients of the language, but differed in some important ways. The modeling projects described in this paper have run in several different versions of the language on several different platforms. For simplicity of the exposition, all models will be described in their re-implemented form in NetLogo.[7]

This paper will describe in detail the evolution of a collection of NetLogo models for exploring the behavior of gases. The models evolved as they were constructed and extended by a high school physics teacher and several high school students. The theory of gases is a classical and central topic in chemistry and physics education – a topic that historically gave rise to the field of statistical mechanics. It has been and still is the subject of considerable writings by leading scientists engaged in intellectual debates replete with subtle arguments and paradoxes. As such, it is considered a difficult topic for graduate students in physics and chemistry. Yet, in this paper, we will “see” average high school students deeply engaged and doing quite sophisticated reasoning about this advanced content.

We now call this collection of models GasLab. The original GasLab model was built, in the connection machine version of StarLogo, by a high school physics teacher involved in the Connected Probability project. He called the model GPCEE (**G**as **P**article **C**ollision **E**xploration **E**nvironment). In the re-implementation of GPCEE for StarLogoT and then for NetLogo, the GPCEE model was renamed Gas-in-a-Box and it is one of an evolving collection of models that constitute GasLab. As part of the Modeling across the Curriculum project (Horwitz et al., 2002), many of the GasLab models have been adapted and have been incorporated as core elements of the (secondary level) Connected Chemistry curriculum (Wilensky, Bruozas and Levy, 2003).

4. THE CREATION OF THE GAS-IN-A-BOX MODEL – HARRY'S STORY

In the context of the Connected Learning projects, students were offered the opportunity to construct NetLogo models of phenomena of interest to them that involved probability and statistics. Harry, a high school physics teacher enrolled in an education class that I was teaching, had long been intrigued by the behavior of a gas in a sealed container. He had learned in college that the gas molecule speeds were distributed according to a famous law, the Maxwell-Boltzmann distribution law. This distribution had a characteristic right-skewed shape. He had taught this law and its associated formula to his own students, but there remained a gap in his understanding – how/why did this particular distribution come about? What kept it stable? To answer these questions, he decided to build (with my help[8]) a multi-agent model of gas molecules in a box.

Harry built his model based on certain classical physics assumptions:

- Gas molecules are modeled as spherical "billiard balls" – in particular, as symmetric and uniform with no vibrational axes.
- Collisions are "elastic" – that is, when particles collide with the sides of the box or with other gas molecules, no energy is lost in the collision, all the energy is diverted back into the kinetic energy of the moving molecules.
- Points of collision between molecules are determined stochastically. Because it can be hard to calculate the exact collision points of many spherical particles, it is reasonable to model the points of collision contact between particles as randomly selected from the surface of the balls.[9]

Harry's model displays a box with a specified number of gas particles randomly distributed inside it. The user can set various parameters for the particles such as: Mass, speed, number. The user can then perform "experiments" with the particles and observe the results in the graphics window,

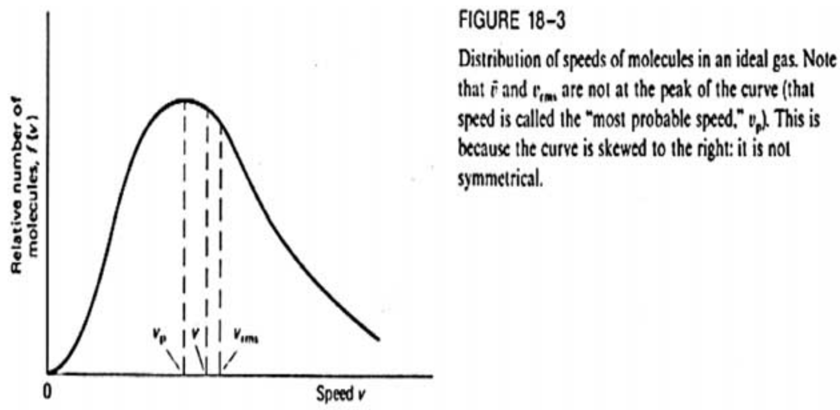

Figure 1. Maxwell-Boltzmann distribution of molecule speeds (illustration from Giancoli, 1984).

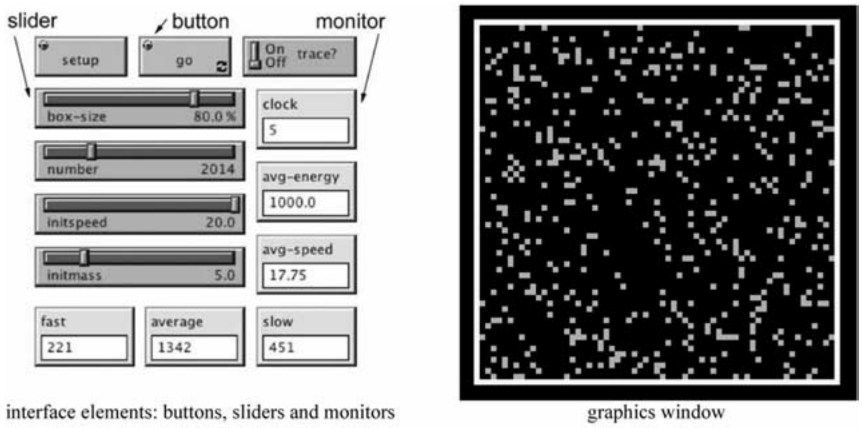

Figure 2. Gas-in-a-Box interface.

or in dynamic plots or in the numeric "monitors" (such as "avg-speed" in Figure 2).

Harry called his program GPCEE though other students subsequently dubbed it "GasLab". Harry's program was a relatively straightforward (though longish) NetLogo program. At its core were three procedures that were executed (in parallel) by each of the particle "agents" in the box:

At each "tick" of the NetLogo "clock":

> **go**: each particle checks for obstacles and, if none are present, moves forward an amount based on its speed variable;
> **bounce**: if the particle detects a wall of the box, it bounces off the wall
> **collide**: if the particle detects another particle in its vicinity, the particles bounce off of each other like billiard balls.

Harry was excited by the fact that the laws of the gas should emerge "automatically" from the simple rules he had written for the particles. He realized that he wouldn't need to program the macro-level gas rules explicitly; they would come "for free" if he wrote the underlying (micro-level) particle rules correctly. He hoped to gain further confidence in the gas laws through this approach – seeing them emerge as the result of the laws of individual particles and not as some mysterious orchestrated properties of the gas.

In one of his first experiments, Harry created a collection of particles of equal mass randomly distributed in the box. He initialized them to start at the same speed but moving in random directions. He kept track of several statistics of the particles on another screen. When looking at this screen, he noticed that one of his statistics, the average speed, was going down. This surprised him. He knew that the overall energy of the system should be constant: Energy was conserved in each of the collisions. After all, he reasoned, the collisions are all elastic, so no energy is lost from the system. Since the number of molecules isn't changing, the average energy or total energy/number of molecules should also be a constant. But energy, he continued, is just proportional to the mass and the square of the speed. Since the mass is constant for all molecules, he concluded that the average speed should also be constant. The puzzle, then, was why did the model output show the average speed to be decreasing? In Harry's words

> The IMPLICATION of what we discovered is that the average length of each of the individual vectors does indeed go down. PICTURE IT!! I visualize little arrows that are getting smaller. These mental vectors are just that. Little 2 (or 3)-dimensional arrows. The move to the scalar is in the calculation of energy (with its v∗∗2 terms.) Doesn't it seem difficult to reconcile the arrows (vectors) collectively getting smaller with a scalar (which is a quantity that for a long time was visualized as a fluid) 'made up' from these little vectors NOT getting less!

Harry was dismayed by this new "bug" and set out to find what "had to" be an error in the model program code. He worked hard to analyze the decline in average speed to see if he could get insight into the nature of the calculation error he was sure was in the program.

But there was no error in the code. After some time unsuccessfully hunting for the bug, Harry decided to print out average energy as well. To his surprise, the average energy stayed constant.

At this point Harry realized that the "bug" must be in his thinking rather than in the code. His reasoning had led him to believe that if average-energy was constant, so would average-speed be constant. But, in this model, average energy was constant but not so average speed. He, thus, felt the need to further explore the behavior of the model in order to understand better how his reasoning could have led him astray. He began his exploration by trying to get a more visual understanding of the gas dynamics. To do so, he decided to color-code the particles according to their speed: Particles are initially colored green; as they speed up, they get colored red; as they slow down, they get colored blue. Soon after starting the model running, Harry observed that there were many more blue particles than red particles. He realized that this was yet another way of thinking about the average-speed problem. If there are more slow (blue) particles than fast (red) ones, then the average speed would indeed have to drop – so this was consistent with the hypothesis that the problem was in his thinking not in the code.

Upon seeing the color asymmetry, Harry realized that, of course, if the particles were to be distributed into the predicted Maxwell-Boltzmann distribution, there would necessarily have to be more slow particles than fast since the distribution was skewed. The color-coding gave him a concrete way of thinking about the asymmetric Maxwell-Boltzmann distribution. He could “see” the distribution: Initially all the particles were green, a uniform symmetric distribution, but as the model developed, there were increasingly more blue particles than red ones, resulting in a skewed asymmetric spread of the distribution.

Even though Harry knew about the asymmetric Maxwell-Boltzmann distribution, he was surprised to see the distribution emerge from the simple rules he had programmed. Since he had programmed the rules, he had faith that this stable distribution does indeed emerge. Harry tried several different initial conditions and all of them resulted in this distribution. He was now starting to believe that this distribution was not the result of a specific set of initial conditions, but that any gas, no matter how the particles’ speeds were initialized, would attain this stable distribution. In this way, the NetLogo model served as an experimental laboratory where the distribution could be “discovered.” This type of experimental laboratory is not easily (if at all) reproducible in a physical experimental setup outside of a computer-based-modeling environment.

But there remained several puzzles for Harry. Though he believed *that* the Maxwell-Boltzmann distribution emerged from his rules, he still did not see *why* they emerged. And he still did not understand how these observations fit with his mathematical knowledge – how could the average speed change when the average energy was constant?

Reflecting on this confusion gave Harry the insight he had originally sought from the GasLab environment. Originally, he had thought that, because gas particles collided with each other randomly, they would be just as likely to speed up as to slow down[10] so the average speed should

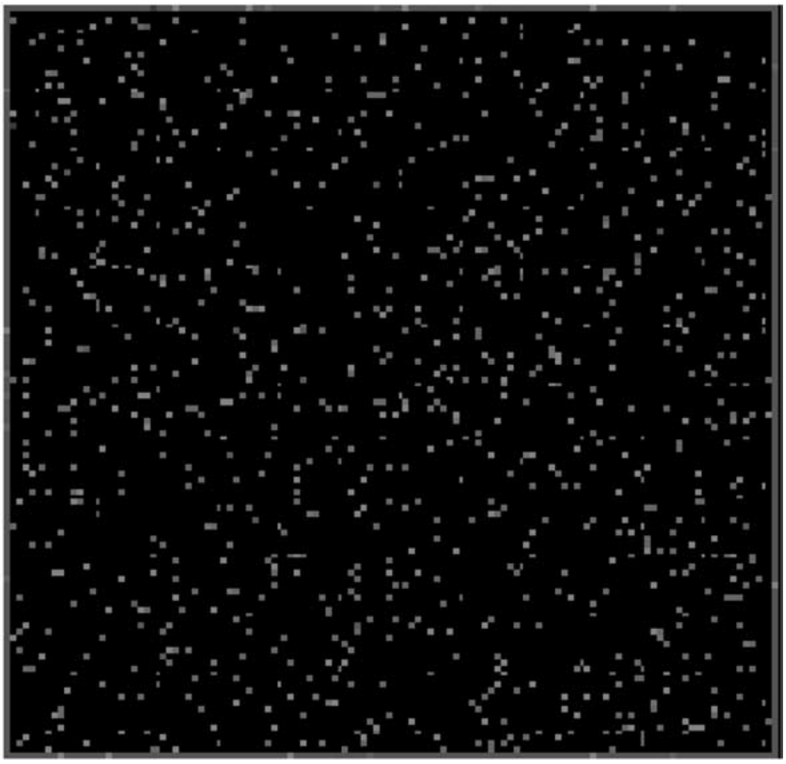

Figure 3. 8000 gas particles after 30 ticks. Faster molecules are red, slower molecules blue, and average-speed molecules are green.

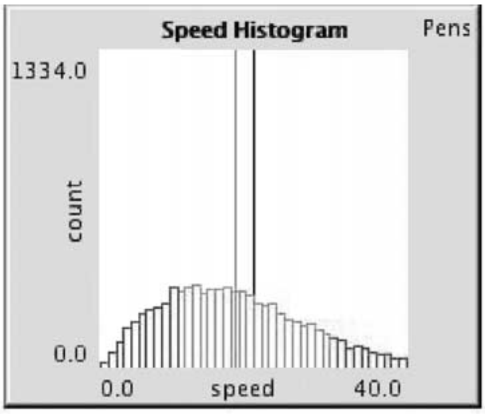

Figure 4. Dynamic histogram of molecule speeds after 30 clock ticks.

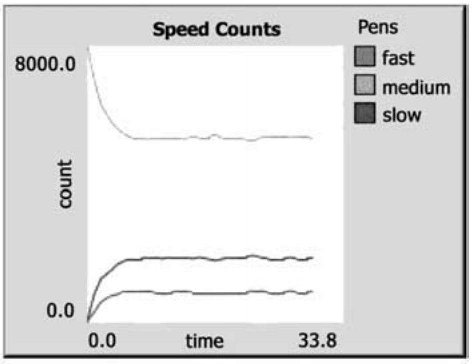

Figure 5. Dynamic plot of fast, slow, and medium speed particles.

stay roughly constant. But now, Harry saw things from the perspective of the whole ensemble. The law of conservation of energy guaranteed, Harry knew, that the overall pool of energy was constant.[11] Although there were many fewer red particles than blue ones, Harry realized that each red particle "stole" a significant amount of energy from this overall pool of energy. The reason: energy is proportional to the square of speed, and the red particles were high speed. Blue particles, in contrast, took much less energy out of the pool. So each red particle need to be "balanced" by more than one blue particle to keep the overall energy constant. In Harry's words:

> There have to be more blue particles. If there were the same number of blues as reds then the overall energy would go up. Let's say 1000 green particles have mass 1 and speed 2, then the overall energy is equal to 2000 [ED − 1/2∗m∗V∗∗2]. If half the greens become red at speed 3 and half become blue at speed 1, then the energy of the reds is 500 ∗ 1/2 ∗ 9 which equals 2250. (Wow, that's already more than the total energy) and the energy of the blues is 500∗1/2∗1 which equals 250. Oh, yeah, I guess I don't need the 500 there [ED – noticing that 500 occurs in both expressions, so it can be factored out], a red is nine times as energetic as a blue so to keep the energy constant we need 9 blues for every red.

Harry was now sure he had discovered the nugget, the crux of why the Boltzmann distribution arose. As particles collided they changed speeds and the energy constraint ensured that there would be more slow particles than fast ones. Yet, he was still puzzled on the "mathematical side". He saw that the greater number of blue particles than red particles ensured that the average speed of the molecules would indeed decrease from the initial average speed of a uniform gas. But, how did this knowledge fit with the classical Newtonian formulas?

Harry had been working on the classical physics equations when he felt sure there was a bug in the NetLogo code. He had been working on them in two different ways and both methods led to the conclusion that the average speed should be constant. What was wrong with his previous reasoning?[12]

In his first method, he had started with the assumption that momentum[13] is conserved inside the box. Since mass is constant, this means the average velocity as a vector is constant. Since the average velocity is constant, he had reasoned that its magnitude, the average speed, had to be constant as well. But, now he saw that this reasoning was faulty, in his words:

> [I] screwed up the mathematics – the magnitude of the average vector is *not* the average speed. The average speed is the average of the magnitudes of the vectors. And the average of the magnitudes is not equal to the magnitude of the average.

In his second method, he began with the assumption that the energy of the ensemble would be constant. This could be written as Σ_i 1/2 mv_i^2 is constant. Factoring out the constant terms, it follows that $\Sigma_i\ v_i^2$ is a constant. From this he had reasoned that the average speed, Σ_i abs(v_i), would also have to be constant. He now saw the error in that mathematics as well. It is not hard to show that if the former sum (corresponding to energy) is constant then the latter sum (corresponding to speed) is maximal under the uniform initial conditions.[14] As the speeds of the particles diverge, the average speed decreases just as he "observed". For a fixed energy, the maximum average speed would be attained when all the speeds were the same – as they were in the initial state. As the system evolves from that initial state, more particles would slow down than would speed up.

Although both these bugs were now obvious to Harry and he felt that they were "embarrassing errors for a physics teacher to make", this confusion between the different kinds of averages was still lurking in the background of his thinking. Once brought to light, it could in principle be readily dispensed with through standard high school algebra. However, the standard mathematical formalism did not cue Harry into seeing his errors. His confusion was brought to the surface, leading to his understanding, through constructing and immersing himself in the Gas-in-a-Box model. In working with the model, it was natural for him to ask questions about the large ensemble and to get experimental and visual feedback. This also enabled Harry to move back and forth between different conceptual levels, the level of the whole ensemble, the gas, and the level of individual molecules.

Harry was now satisfied that the average speed of the ensemble would indeed decrease from its initial uniform average. The above reasoning relieved his concerns about how such an asymmetric ensemble could be stable. But it had answered his question only at the level of the ensemble. What was going on at the level of individual collisions? Why were collisions more likely to lead to slow particles than fast ones? This led him to

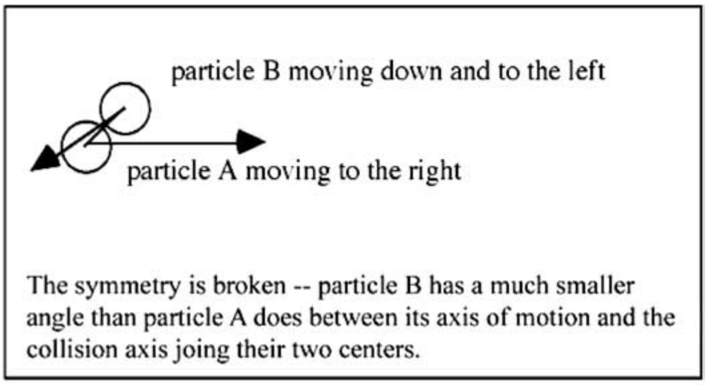

Figure 6. Broken symmetry leads to changing speeds.

conduct further investigations into the connection between the micro- and macro-views of the particle ensemble.

Harry was led inexorably to the question: Why would the particle speeds spread out from their initial uniform speed? Indeed, why do the particles change speed at all? Harry knew that the particles change their speeds when colliding with other particles, but he wondered: “But the collisions between particles are completely symmetric – why does one particle change speed more than the other?” To answer this question, Harry conducted further modeling experiments that focused on only two particles that repeatedly collided in fixed trajectories. After seeing two particles collide at the same angle again and again, but emerging at different angles each time, he remembered that “randomness was going on here”. The particles were“ choosing” random points on their surface to collide, so they did not behave the same way each time. By experimentally varying the collision points, he observed that the average speed of the two particles did not usually stay constant. Indeed, it remained constant only when the particles collided head-on.

It was not long from this realization to the discovery of the broken symmetry: “when particles collide, their trajectories may not be symmetrical with respect to their collision axis. The apparent symmetry of the situation is broken when the particles do not collide head-on – that is, when their directions of motion do not have the same relative angle to the line that connects their centers” (Figure 6).

Harry then went on to do the standard Newtonian physics calculations that confirmed this experimental result. In a one-dimensional world, he concluded, all collisions would be head on and, thus, average speed would stay constant; in a multi-dimensional world, collisions cause particle speed distributions to become non-uniform and this leads inevitably to the preponderance of slower particles and the characteristic asymmetric distribution.

Harry had now adopted many different views of the gas and used many different methods to explain the asymmetry of the particle speed distribution. Through connecting the macro-view of the particle ensemble with the micro-view of the individual particle collisions, he had come to understand both levels of description in a deeper way. Through connecting the mathematical formalism to his observations of colored particle distributions, he caught errors he had made on the "mathematical side" and, more importantly, anchored the formalism in visual perception and intuition. Harry felt he had gained great explanatory power through this connection of the micro- and macro-view. This connection was made feasible through the support offered by the NetLogo modeling language.

When asked what he had learned from the experience of building the Gas-in-a-Box model, Harry made one more trenchant observation. He had found that the average speed of the gas molecules was not constant. Upon reflection, he had a "meta-cognitive" realization:

> Of course the average speed is not constant. If it were constant, I'd have known about it. It isn't easy to be a constant and that's why we have named laws when we find constants or invariants. The law of conservation of energy guarantees that the energy of the gas is a constant. We do not have a law of conservation of speed.

In saying "it isn't easy to be a constant", Harry was understanding the concept of energy in a new way. He saw that energy could be seen as a statistical measure of a particle ensemble and that it was a special such measure, a measure that characterized the gas across all its changes. He saw that there could be many statistical measures that characterize an ensemble – each of them could lay claim to being a kind of "average", that is a characteristic measure of the ensemble. The idea of "average" is thus seen to be another method for summarizing the behavior of an ensemble. Different averages are convenient for different purposes. Each has certain advantages and disadvantages, certain features that it summarizes well and others that it doesn't. Which average we choose or construct depends on what aspect of the data is important to us. Energy, he now saw, was a special such average – a measure that characterized *invariantly* a fundamental aspect of the collection of particles in a box. Of the many measures we can use to characterize a gas, only a very few had this invariant property. Speed, as he observes, does not. Harry thus came to see energy, no longer as a mysteriously chosen formula, but rather as a measure selected by scientists for human reasons[15] – selected from the set of possible measures for its utility in unchangingly describing the gas as its molecules moved and collided.

5. CREATION OF THE GASLAB TOOLKIT – EXTENSIBLE MODELS

After Harry finished working with the Gas-in-a-Box model, I decided to test the model with students who had not been involved in its development. I contacted a local high school and arranged to meet three hours a week for several weeks with a few juniors and seniors taking introductory physics. The group met with me when we could work it into our schedules, sometimes during free periods and sometimes before or after school. The group was somewhat fluid, consisting of three regular members with 3–4 others sometimes dropping in. The students who chose to be involved did so out of interest. Their teacher described the 3 regular members as "average to slightly above average" physics students in his (non-honors) class. I introduced the students to the Gas-in-a-Box model, showed them how to run the model and how to change elementary parameters of the model. I asked them to begin by just "playing" with the model and talking to me about what they observed. I describe below these students' experience with GasLab. I have introduced GasLab to dozens of groups of students (high school and collegiate) since that time. While the details of their explorations are quite different in each case, the overall character of the model-based inquiry is typified by the story related below.

The students worked as a group, one of them "driving" the model from the keyboard with others suggesting experiments to try. One of the first suggested experiments was to put all of the particles in the center of the box.[16] This led to an aesthetically pleasing result as the gas "exploded" in rings of color, a red ring on the outside, with a nested green ring and a blue ring innermost.

The students soon hit upon the same initial experiment that had stimulated Harry. They started with a uniform distribution of 8000 green particles and immediately wondered at the preponderance of blue particles over red particles as the simulation unfolded. Over the next week, they went through much of the same reasoning that Harry had gone through connecting the energy economy of the gas particle ensemble with the speed distribution of the particles.

But these students were not as motivated by this particular question as Harry had been. One student, Albert, became very excited by the idea that the laws of physics would emerge from the model rules:

> What's really cool is that this is it. If you just let this thing run then it'll act just like a real gas. You just have to start it out right and it'll do the right thing forever. We could run experiments on the computer and the formulas we learned would come out.

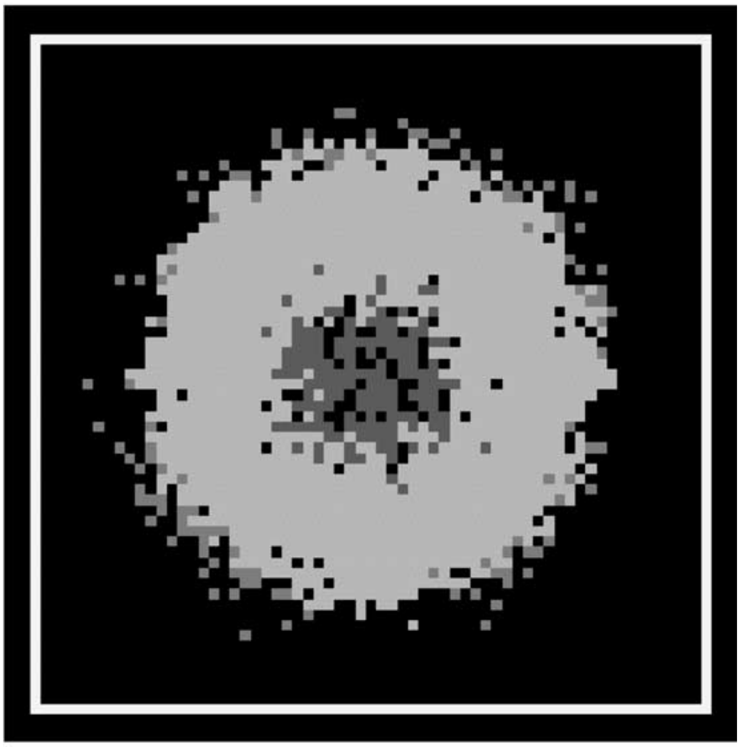

Figure 7. Gas molecules “explode” from the center of the box.

Albert went on to suggest that since this was a fidelous simulation (“a real gas”), they could verify the ideal gas laws for the model. The group decided to verify Boyle’s law that changing the volume of the box would lead to an inverse proportional change in the pressure of the gas.

Now the group was faced with creating an experiment that would test whether Boyle’s law obtained in the GasLab model. Tania made a suggestion:

> We could make the top of the box move down like a piston. We’ll measure the pressure when the piston is all the way up. Then we’ll let it fall to half way down and measure the pressure again. The pressure should double when the piston is half way down.

The group agreed that this was a reasonable methodology, but then were stopped short by Isaac who asked: “How do we measure the pressure”? This question was followed by a substantial pause. They were used to being given instruments to measure physical magnitudes, such as pressure, a black box that they could just read out a number from. As Albert said for the group: “We have to invent a pressure-measure, a way of saying what the pressure is in terms of the particles". The group pondered this question for the next several days. Tania suggested the first operational measure:

> we could have the sides of the box store how many particles hit them at each tick. The total number of particles hitting the sides of the box at each tick is our measure of pressure.

They programmed this measure of pressure into the model. Lots of discussion ensued as to what units this measure of pressure represented. At long last, they agreed that they did not really care what the units were. All they needed to know, in order to verify Boyle's law, is that the measure would double, so a scale factor (due to units) would not affect the result of the experiment.

They created a "monitor" that would display the pressure in the box and ran the model. To their dismay, the pressure in the box fluctuated wildly. Tania was quick to point out the problem:

> We only have 8000 particles in the box. Real boxes full of gas have many more particles in them. So the box is getting hit a lot less times at each tick than it should be. I think what's happening is that the number of particles isn't big enough to make it come out even.

Persuaded by this seat-of-the-pants "law of large numbers" argument, they made an adjustment to the pressure measuring code. They calculated the number of collisions at each tick over a number of ticks, then averaged them. Trial and error simulations varying the averaging time interval convinced them that averaging over ten ticks led to a sufficiently stable measure of pressure.[17]

Now that they had a stable pressure gauge, they were ready to construct the piston and run the experiment. But, here again, they ran into conceptual difficulties. How was the piston to interact with the particles? Were they to model it as a large massive particle that collided with the particles? In that case, how massive should it be? And, if they did it that way, wouldn't it affect the pressure in the box in a non-uniform way? As Albert said:

> If we do the piston, then the North-South pressure in the box will be greater than the East-West pressure, that doesn't seem right. Shouldn't the pressure in the box stay even?

This issue was discussed, argued and experimented on for several hours. It was at this point that Tania suggested another approach.

> I'm confused by the effect the piston is supposed to have on the particles. I have an idea. Why don't we start the particles out in half the box, then release the "lid" and let them spread out into the whole box. If we do that, we won't have to think about pistons and we can just see if the pressure decreases in half.

Figure 8. Box with lid down: volume = 1200 box with lifted lid: volume = 2400 lifting box lid proportionally reduces pressure.

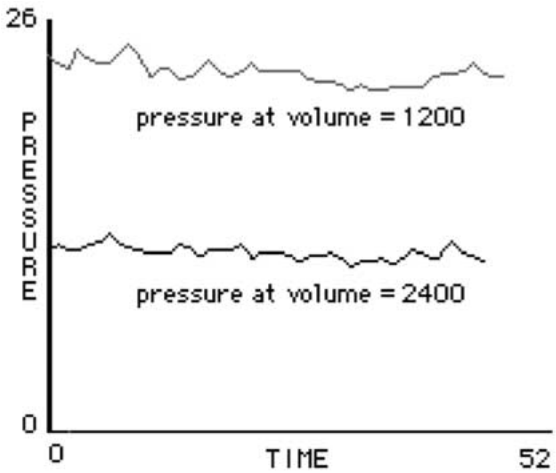

Figure 9. Plot of pressure as measured in the box at two different volumes.

The group agreed that this was a promising approach and quickly implemented this code. They were now able to run the experiment that they hoped would confirm Boyle's law. Indeed their experiment worked as they hoped. When they lifted the lid so that the box had double the volume, the pressure in the box did indeed drop in half.

This confirming result could have led to an unfortunate acceptance of Tania's measure of pressure as accurate. Indeed, if the experiment had been reversed, that is Boyle's law was taken as given, they would not have been able to disconfirm this pressure measure by experimental results. Fortunately, there were other reasons to question Tania's pressure measure. The group had by now developed the habit of making sense of macro-level quantities such as pressure in terms of the micro-level interactions. Intuitively, if you have more massive or faster particles, the force they impart to the box is greater. The students tested this intuition against the model by experimenting with a single particle in a box and observed that (with the pressure measure they had implemented) the pressure did not change as they changed the mass of the particle. This conflict of the model with their intuition led them to revise their pressure measure, in effect to reinvent a classical measure of conventional physics – momentum transfer to the sides of the box per unit time.[18] Their escapade with Tania's definition, however, did yield insights as to the essential mechanisms behind Boyle's law. As Tania later said:

> I guess for Boyle's law to work, all that matters is how dense the molecules are in the box. With more space [ED: more room in the box] they're less likely to collide [ED: with the sides of the box per time unit] so the pressure drops.

There is another incident of note surrounding the Boyle's law experiment. A week or so after completing the experiment, Isaac ran the model again with all particles initialized to be at the center of the box. While watching his favorite "explosion", Isaac noted that the gas pressure registered 0! Quickly, he realized that that was a consequence of their definition of pressure – no particles were colliding with the sides of the box. This result didn't seem right to Isaac and led him to ask the group if they should revise their concept of pressure yet again. Argumentation ensued as to "whether a gas had internal pressure without any box to measure it". They realized that the experiment in question was not feasible in a real experimental setting, but nonetheless, it did seem that there should be a theoretical answer to the question. Isaac suggested various ingenious solutions[19] to the problem, but in the end, the group did not alter their pressure gauge. The ingenious solutions were more difficult to implement and their current gauge seemed to be adequate to the experiments they were conducting.[20]

Another noteworthy development related to the emergence of the Boltzmann distribution discussed in the previous section. Albert came in one day all excited about an insight he had had. The gas molecules, he said, can be thought of as probabilistic elements, sort of like dice. They can randomly go faster or slower. But while there is no real limit to how fast they can go,[21] their speed is bounded below by zero speed. It's as if particles were conducting a random walk on the right half plane but there was a wall on the y-axis. Albert saw that this constrained random walk would have to produce a right-skewed distribution. I challenged him to go further: (a) Could he construct a NetLogo model to prove his theory? (b) Could he determine what particular probability constraints would produce a strict Boltzmann distribution? (c) Could he find other seemingly unrelated phenomena that satisfied the same formal constraints and thus would also produce a Boltzmann distribution? Albert and his fellow students were up to these challenges.

These students (and subsequent groups of students) have conducted many more experiments with the Gas-in-a-Box model. As they revised and extended the model, they created a set of models that has since been expanded and revised into the toolkit we now call GasLab. The set of extensions of the original Gas-in-a-Box model is impressive in its scope and depth. Among the many extensions they tried were: Heating and cooling the gas, introducing gravity into the model (and a very tall box) and observing atmospheric pressure and density, modeling the diffusion of two gases, allowing the top to be porous and seeing evaporation, relaxing elasticity constraints and looking for phase transitions, introducing vibrations into the container and measuring sound density waves, and allowing heat to escape from the box into the surrounding container. They also reinvented various well-known thought experiments of statistical mechanics related to Maxwell's demon and second law considerations.[22]

5.1. *Statistical Mechanics for Secondary School*

Over the course of several weeks, these high school students "covered" much of the territory of collegiate statistical mechanics and thermal physics and their understanding of it was deeply grounded in both (a) their intuitive understandings gained from their inquiry experience with the models and (b) the causal and explanatory relationships between the micro-behavior of the gas molecules and the macro-features of the gas.

It is worth noticing that this is a remarkable achievement. As has been noted, the territory of statistical mechanics is a difficult and confusing one even for graduate students in physics. These graduate students have typically had several years of physics instruction at an undergraduate level and intensive instruction at a graduate level prior to undertaking this material. Professional physicists, too, find this material challenging and requiring of careful and subtle argumentation. How, then, is it possible for average high school students with virtually no physics background to make significant headway?

We start our reply to this provocative question with some caveats. Clearly, there is much of the territory of statistical mechanics these students did not cover. In particular, they, of course, did not cover the analytic differential and statistical techniques that make up a large part of statistical mechanics instruction. What is interesting is that despite not being able to use the analytic techniques, these high school students were able to deeply engage with the ideas, models and thought experiments that are central to the study of statistical mechanics. The multi-agent modeling environment together with the GasLab toolkit afford more direct engagement with these core ideas and experiments – enabling a much larger and younger segment of the population to appreciate the beauty and explanatory power of these ideas. By enabling students to shift perspectives from the micro-level of the particles to the macro-level of the gas and to vary properties at each level and immediately see the consequences at another level, GasLab fosters a more general shift in student perspective, a perspective of emergence, through which they come to fluently see the properties of the gas as causally arising from the properties and interactions of individual gas molecules. This new perspective serves as the substrate for an emerging literacy in dynamic systems, an ability to decode large-scale patterns into their component elements and interactions and to understand how a change in the properties of the elements could affect the behavior of the system. To be more fully literate in this domain, they would eventually need to master some of the analytic and symbolic techniques that would enable them to do more exact analyses and predictions of system behavior. Only a small number of students will likely choose to achieve this higher level of literacy, but, even for those, it seems likely that when they encounter the mathematical techniques of statistical mechanics, they will make greater sense as they will be situated as augmentations of their qualitative understandings of these systems.

GasLab provides learners with a set of tools for exploring the behavior of an ensemble of micro-elements. Through running, extending, and creating GasLab models, these learners were able to develop strong intuitions about the behavior of the gas at the macro level (as an ensemble gas entity) and its connections to the micro level (the individual gas molecule). In a typical high school physics classroom, learners usually address these levels at different times. When attending to the micro level, the focus is, typically, on the exact calculation of the trajectories of two colliding particles. When attending to the macro level, the focus is on “summary statistics” such as pressure, temperature, and energy. Yet, it is in the connection between these two primary levels of description that the great explanatory power resides.

Two major factors enable students using GasLab to make the connection between these levels – the replacement of symbolic computation with simulated experimentation and the replacement of “black-box” summary statistics with learner-constructed summary statistics. The traditional secondary physics curriculum segregates the micro- and macro-levels of description because the mathematics required to meaningfully connect them is thought to be out of reach of high school students. In the GasLab modeling toolkit, the formal mathematical techniques can be replaced with more concrete experimentation with simulated particles. This experimentation enables learners to get immediate feedback on their theories and conjectures. In traditional curricula learners are typically handed concepts such as pressure as "received" physics knowledge. The concept (and its associated defining formula) is, thus, for the learner, a "device" built by an expert, which the learner can neither inspect nor question. Learners do not come to see that this concept represents a summary statistic – a way of averaging or aggregating the behavior of many individual particles (see e.g., Wilensky, 1997b). Most fundamentally, the learner has no access to the design space of possibilities from which this particular summary statistic was selected. In the GasLab context, learners must construct their own summary statistics. As a result, the traditional pressure measure is seen to be one way of summarizing the effect of the gas molecules on the box, one way to build a gauge. The activity of designing a pressure measure is an activity of doing physics, not absorbing an expert's "dead" physics.

These two factors, the ability to act on the model and to "see" its reactions and the ability to create interpretations of the model in the form of new computational objects which, in turn, can be acted upon make a significant difference in the kinds of understandings students can construct of the behavior of gas molecule ensembles. Through such activities, the students came to understand the gas as a concrete entity,[23] much in the same way they experience physical entities outside the computer. Through engaging with GasLab, high-school students have access to the powerful ideas and explanations of statistical thermal physics. These constructive modeling and model-based reasoning activities can provide students a powerful way of apprehending the physics and chemistry of gases – one that eludes even many professional scientists who learned this content in a traditional manner.[24]

6. IMPLICATIONS FOR THE PEDAGOGY OF MODELING

Despite the rapid rate of infiltration of computer-based modeling and dynamics systems theory into scientific research and into popular culture, computer-based modeling has only slowly begun to impact education communities. While computer-based models are increasingly used in the service of pedagogic ends (Buldyrev et al., 1994; Chen and Stroup, 1993; Doerr, 1996; Feurzeig, 1989; Horwitz, 1989; Horwitz et al., 1994; Jackson et al., 1996; Mandinach and Cline, 1994; Mellar et al., 1994; Roberts et al., 1983; Repenning, 1994; Shore et al., 1992; Smith et al., 1994; White and Frederiksen, 1998; Wilensky, 1997; Wilensky and Resnick, 1999; Wilensky and Reisman, in press; Stieff and Wilensky, 2003), there remains significant lack of consensus about the proper role of modeling within the curriculum.

6.1. *Model Construction Versus Model Use*

One tension that is felt is between students using already-constructed models of phenomena versus students constructing their own models to describe phenomena. At one extreme is the use of pre-constructed models purely for demonstration of phenomena. This use of modeling employs the computer to animate and dynamically display the structures and processes that describe the phenomena, much the way playing an animated movie might do so. At the other extreme, learners are involved in constructing their own models of phenomena *de nova*. Between these extremes are other kinds of modeling activities: One of particular interest is student use of pre-constructed models as investigative tools for model-based inquiry – activities that may involve learner modification and extension of the initial models provided to them. Here, students are given starting models but are also involved in model design and development.

For the use of models to provide classroom animations, I employ the term “demonstration modeling”. While such demonstration models can be visually striking, they are not very different from viewing a movie of the phenomenon in question. The computational medium is being used merely for delivery. From a constructivist point of view, this delivery model is unlikely to lead to deep learning, as it does not engage with the learner’s point of entry into the phenomena to-be-understood. Nor does this approach take advantage of the computer’s interactivity to give the learner a chance to probe the model and get the feedback necessary to construct mental models of the phenomena observed.

Constructivists might be happier with the “from scratch” modeling activity, as it requires the learner to start where she is at and interact with the modeling primitives to construct a model of the phenomenon. That special breed of constructivist called constructionists (Papert, 1991) would argue that this externalized construction process is the ideal way to engage learners in constructing robust mental models. The learner is actively engaged in formulating a question, formulating tentative answers to her question and through an iterative process of reformulation and debugging, arriving at a theory of how to answer the question instantiated in the model. This process is an act of doing and constructing mathematics and science instead of viewing the results of an expert’s having done the mathematics and science and handing it off to the learner. On the epistemological side, this lesson that mathematics and science are ongoing activities in which ordinary learners can be creative participants is an important meta-lesson of the modeling activity. These considerations can be summarized in the table given below:

	Model observation (demonstration models)	Model construction (model based inquiry)
Student's role	Passive	Active
Student's activity	Viewing "received" mathematics	Constructing mathematics
Communication	Transmitting/receiving ideas	Expressing ideas
Student goal	View output of mathematical description	Symbolize, express, and refine mathematical description
Source of question	The expert	The learner
Source of solution	The expert	The learner (tentative)
Learning style	Single step	Through debugging and successive refinement
Design of learning parameters (content, sequence)	Experts must anticipate relevant parameters for learning	Learner can construct parameters relevant to their learning
Feedback	Limited	Constant, immediate, specific

In reply to these powerful arguments offered on behalf of model construction, an argument advanced on the side of demonstration modeling is that the content to be learned is placed immediately and directly to the attention of the learner. In contrast, in the process of constructing a model, the learner is diverted into the intricacies of the modeling language itself and diverted away from the content to be learned. Since there can be quite a bit of overhead associated with learning the modeling language, the model construction approach could be seen as inefficient. Moreover, there is skepticism as to whether students who are not already mathematically and scientifically sophisticated can acquire the knowledge and skills of model design and construction.

6.2. *Selecting the Appropriate "Size" of Modeling Primitives*

Like most tensions, this tension between model observation and model construction is not really dichotomous. There are many intermediate states between the two extremes. Demonstration models can be given changeable parameters that users can tune to explore the effect of modifying initial settings on the behavior of the model. If there are large numbers of such parameters, as in the popular Maxis simulation software packages (1992a, 1992b), the parameter space can be quite vast in its exploratory potential. This takes demonstration models several steps in the direction of model construction. On the other hand, even the most "from scratch" modeling language must contain primitive elements. These primitive elements remain black boxes, used for their effect but not constructed by the modeler. Not too many constructionist modelers would advocate building the modeling elements from the binary digits, let alone building the hardware that supports the modeling language. The latter can serve as an absurd reductio of the "from scratch" label. So, even the die-hard constructionist modelers concede that not all pieces of the model need be constructed – some can be simply handed off.

As I place myself squarely in the constructionist camp, the challenge for us is to construct toolkits that contain an appropriate set of primitives. One important dimension to attend to in selecting a set of primitives, is the level or grain-size of the primitives. In constructing a modeling language, it is critical to design primitives not so large-scale and clunky that they can only be put together in a few possible ways. If we fail at that task, we have essentially reverted to the demonstration modeling activity since the learners construct a model from these large pieces that can only be put together in our pre-conceived way. To use a physical analogy, we have not done well in designing a dinosaur modeling kit if we provide the modeler with three pieces, a t-rex head, body and tail. On the other hand, we must design our primitives so that they are not so "small" that putting them together is seen by learners as far removed from the objects they want to model. If we fail at that task, learners will be focused at an inappropriate level of detail and so will learn more about the modeling pieces than the content domain to be modeled and, furthermore, the effort involved in construction becomes too great. To reuse the physical analogy, designing a dinosaur modeling kit to consist of large numbers of small metal hardware units may make constructing many different kinds of dinosaurs possible, but it will be tedious and removed from the functional issues of dinosaur physiology that form the relevant content domain.[25]

This places modeling language designers face to face with the challenge of fine-tuning the grain-size of the primitive modeling elements to be given to learners. Modeling language designers who choose to make their primitive elements on the large side, we call demonstration modeling designers, whereas those that tend to keep their primitives small, we call constructionist modeling designers. Demonstration modeling designers have no choice but to make the pieces from which the models are built semantically interpretable from within the model content domain. Constructionist modeling designers, though, can make the underlying model elements content neutral,[26,27] thus creating a modeling language that is general purpose, or they can choose modeling elements that have semantic interpretation in a chosen content domain, thus creating a modeling toolkit for that content domain. We note that in speaking of primitive elements, we can speak more broadly than just the primitive operations of the modeling language. The above discussion could refer to the primitive tasks included in the graphical user interface or even more broadly the primitive student activities afforded by the pedagogy.[28]

6.3. *General Purpose vs. Content Domain Modeling Languages*

Both of these choices, to build content domain modeling languages or to build general purpose modeling languages, can lead to powerful modeling activities for learners. The advantage of the content domain modeling language is that learners can enter more directly into content issues of the domain (issues that will seem more familiar to them and to their teachers, and possibly relevant to some desired curricular content). A disadvantage is that the primitive elements of the language, which describe important domain content, are opaque to the learner. Another disadvantage is that the language can be used only for its specific content domain, though that disadvantage may be nullified by designing a sufficiently broad class of such content domain modeling languages. The advantage of the content-neutral primitives is that all content domain structures, since they are made up of the general-purpose elements, are inspectable, modifiable, and constructible by the learner. The disadvantage is that the learner must master a general-purpose syntax before being able to make headway on the domain content. What we'd like is a way for learners to be able to begin at the level of domain content, but not be limited to unmodifiable black-box primitives at that level.

In the Connected Learning projects, the solution we have found to this dilemma is to build so-called "extensible models" (Wilensky, 1997). In the spirit of Eisenberg's programmable applications (Eisenberg, 1991), these are content-specific models that are built using the general purpose NetLogo modeling language. This enables learners to begin their investigations at the level of the content. Like the group of high schoolers described in the earlier section of this paper, they begin by inspecting a pre-built model such as Gas-in-a-Box. Using pre-built parameter modulators (e.g., sliders on the interface), they can adjust the values of parameters of the model such as mass, speed, and location of the particles and readily conduct experiments at the level of the content domain of ideal gases. But, since the Gas-in-a-Box model is built in NetLogo, the students have access to the workings of the model. They can look "under the hood" and see how the particle collisions are modeled. Furthermore, they can modify the primitives, investigating what might happen if, for example, collisions are not elastic. Lastly, students can introduce new concepts, such as pressure, as primitive elements of the model and conduct experiments on these new elements.

This extensible modeling approach enables learners to dive right into the model content, but places no ceiling on where they can take the model. Mastering the general purpose modeling language is not required at the beginning of the activity, but happens gradually as learners seek to explain their experiments and modify and extend the capabilities of the model so as to suit their individual inquiry processes.

When engaged in classroom modeling, the pedagogy used in the Connected Learning projects has four basic stages: In the first stage, the teacher presents a “seed” model to the whole class. Typically, the seed model is a simply coded model with relatively simple rules. The model is projected on a screen so the whole class can view it. The teacher engages the class in discussion as to what is going on with the model. Why are they observing that particular behavior? How would it be different if model parameters were changed Is this model a good model of the phenomenon it is meant to model? In the second stage, students run the model (either singly or in small groups) on individual computers. Here they engage in systematic “search” of the parameter space of the model. They make hypotheses and test them against the model behavior often producing lab reports of their results. In the third stage, each modeler (or group) proposes an extension to the model and implements that extension in the NetLogo language. Modelers that start with Gas-in-a-Box, for example, might try to build a pressure gauge, a piston, a gravity mechanism or heating/cooling plates. The results of this model extension stage are often quite dramatic, and the extended models are added to the project’s library of extensible models and made available for others to work with as new “seed” models. In the final stage, students are asked to propose a phenomenon to be modeled and to build the model “from scratch” using the NetLogo modeling primitives.

6.4. *Phenomena-Based vs. Exploratory Modeling*

When learners are engaged in creating their own models, two primary avenues are available. A modeler can choose a phenomenon of interest in the world and attempt do duplicate that phenomenon on the screen. Or, a modeler can start with the primitives of the language and explore the possible effects of different combinations of rules sets. The first kind of modeling, which I call phenomena-based modeling (Wilensky, 1997b; Resnick and Wilensky, 1998) is also sometimes called backwards modeling (Wilensky, 1997b) because the modeler is engaged in going backwards from the known phenomenon to a set of underlying rules that might generate that phenomenon. In the GasLab example, Harry knew about the Maxwell-Boltzmann distribution and tried creating rules which he hoped would duplicate this distribution. In this specific case, Harry did not have to discover the rules himself because he also knew the fundamental rules of Newtonian mechanics that would lead to the M-B distribution. The group of students who worked on modeling Boyle's law came closer to pure phenomena-based modeling as they tried to figure out the "rules" for measuring pressure. Phenomena-based modeling can be quite challenging, as discovering the underlying rule-sets that might generate a phenomenon is inherently difficult – it is a fundamental activity of practicing scientists. In practice, most GasLab modelers mixed some knowledge of what the rules were supposed to be with adjustments to those rules when the desired phenomenon did not appear.

The second kind of modeling is sometimes called "forwards" modeling (Wilensky, 1997b) because modelers start with a set of rules and try to work forwards from these rules to some, as yet, unknown phenomenon. The patterns that arise from playing with and adjusting the rule sets might suggest a familiar phenomenon and thus serve as the seed for a model of that phenomenon.[29]

6.5. *New Forms of Symbolization*

In a sense, modeling languages are always designed for phenomena-based modeling. However, once such a language exists, it also becomes a medium of expression in its own right. Just as, we might speculate, natural languages originally developed to communicate about real world objects and relations, but, once they were sufficiently mature, they were also used for constructing new language objects and relations. Shakespeare's Hamlet, if you will, is a new linguistic phenomenon built out of the rules of natural language. Similarly, learners can explore sets of rules and primitives of a modeling language to see what kinds of emergent effects may arise from their rules. In some cases, this exploratory modeling may lead to emergent behavior which resembles some real world phenomenon and then phenomena-based modeling resumes. In other cases, though the emergent behavior may not strongly connect with real world phenomena, the resulting objects or behaviors can be conceptually interesting or beautiful in themselves. In these latter cases, in effect, the modelers have created new phenomena, objects of study that can be viewed as new kinds of mathematical objects–objects expressed in the new form of symbolization afforded by the modeling language.

6.6. *Aggregate vs. Object-Based Modeling*

In a previous section, we discussed the selection of modeling language primitives in terms of size and content-ladenness. Yet another distinction is in the conceptual description of the fundamental modeling unit. To date, modeling languages can be divided into two kinds: so-called “aggregate” modeling engines (e.g., STELLA (Richmond and Peterson, 1990), Link-It (Ogborn, 1994), VenSim (Ventana, 2002), Model-It (Jackson et al., 1996)) and “agent-based” modeling languages (e.g., NetLogo (Wilensky, 1999), StarLogo (Resnick, 1994); StarLogoT (Wilensky, 1997), Agent-sheets (Repenning, 1993), Cocoa (Smith et al., 1994), Swarm (Langton and Burkhardt, 1997), ASCAPE (Parker, 2001) and Repast (Collier, 2000)). Aggregate modeling languages use “accumulations” and “flows” as their fundamental modeling units. For example, a changing population of rabbits might be modeled as an “accumulation” (like water accumulated in a sink) with rabbit birth rates as a “flow” into the population and rabbit death rates as a flow out (like flows of water into and out of a sink). Other populations or dynamics – e.g., the presence of “accumulations” of predators – could affect these flows. This aggregate based approach essentially borrows the conceptual units, its parsing of the world, from the mathematics of differential equations.

The second kind of tool enables the user to model systems directly at the level of the individual elements of the system. For example, our rabbit population could be rendered as a collection of individual rabbits each of which has associated probabilities of reproducing or dying. The agent-based approach has the advantage of being a natural entry point for learners. It is generally easier to generate rules for individual rabbits than to describe the flows of rabbit populations. This is because the learners can literally see the rabbit-agents and can control an individual rabbit’s behavior. In NetLogo, for example, students think about the actions and interactions of individual agents or creatures. NetLogo models describe how individual creatures (not overall populations) behave. Thinking in terms of individual creatures seems far more intuitive, particularly for the mathematically uninitiated. Students can imagine themselves as individual rabbits and think about what they might do. In this way, NetLogo enables learners to “dive into” the model (Ackermann 1996) and make use of what Papert (1980) calls “syntonic” knowledge about their bodies. By observing the dynamics at the level of the individual creatures, rather than at the aggregate level of population densities, students can more easily think about and understand the population dynamics that arise.[30]

There are now some very good aggregate computer modeling languages – such as STELLA (Richmond and Peterson 1990) and Model-It (Jackson et al., 1996). These aggregate models are very useful – and superior to agent-based models in some contexts, especially when the output of the model needs to be expressed algebraically and analyzed using standard mathematical methods. They eliminate one "burden" of differential equations – the need to manipulate symbols – focusing, instead on more qualitative and graphical descriptions of changing dynamics. But, conceptually, they still rely on the differential equation epistemology of aggregate quantities.

Some refer to agent-based models as "true computational models" (Wilensky and Resnick, 1999) since they leverage computational media in a fundamentally more powerful way than most computer-based modeling tools. Whereas most tools simply translate traditional mathematical models to the computer (e.g., numerically solving traditional differential-equation representations), agent-based languages such as NetLogo provide new representations that are tailored explicitly for the computer. Too often, scientists and educators see traditional differential-equation models as the *only* approach to modeling. As a result, many students (particularly students alienated by traditional classroom mathematics) view modeling as a difficult or uninteresting activity. What is needed is a more pluralistic approach, recognizing that there are many different approaches to modeling, each with its own strengths and weaknesses. A major challenge is to develop a better understanding of when to use which approach, and why. A promising new project (Wilensky and Stroup, 1999) attempts to integrate these two types of modeling environments and enable users to go back and forth between these two types of modeling and reasoning.

6.7. *Concreteness vs. Formalism*

Critiques of computer-modeling have come from both sides of the "concrete-abstract divide". Some critics have worried that the models are not sufficiently formal and rigorous and others that they are not sufficiently concrete and real-world.

On the one hand, some mathematicians and scientists have criticized computer models as insufficiently rigorous (see Tymoczko, 1979). As discussed in the previous section, it is not always easy, for example, to get a hold of the outputs of a NetLogo model in a form that is readily amenable to symbolic manipulation. Moreover, there is as yet no formal methodology for verifying the results of a model run. Even in highly constrained domains, there is not a formal verification procedure for guaranteeing the results of a computer-based experiment; much less any guarantee that the underlying assumptions of the modeler are accurate. Computational models, in general, are subject to numerical inaccuracies dictated by finite precision. Agent-based models, in particular, are also vulnerable to assumptions involved in transforming a (perhaps) continuous world into a discrete model. These difficulties lead many formalists to worry about the accuracy, utility, and especially the generality of a model-based inquiry approach (Wilensky, 1996). These critiques raise valid concerns, concerns that must be reflected upon as an integral part of the modeling activity. As we recall, Harry had to struggle with just such an issue when he was unsure whether the drop in the average speed of the gas particles was due to a bug in his model code or due to a “bug” in his thinking. It is an inherent part of the computer modeling activity to go back and forth between questioning the model’s faithfulness to the modeler’s intent (e.g., code bugs) and questioning the modeler’s expectations for the emergent behavior (e.g., bugs in the model rules). Though the formalist critic may not admit it, these limitations are endemic to *modeling* per se – including formal methods such as differential equations. Only a small set of the space of differential equations is amenable to analytic solution. Most perturbations of those equations lead to equations that can only be solved through numerical techniques. The game for formal modeling, then, becomes trying to find solvable differential equations that can be said to map onto real world phenomena. Needless to say, this usually leads to significant simplifications and idealizations of the situation. The classic Lotka-Volterra equations (Lotka, 1925), for example, which purport to describe the oscillations in predator/prey populations assume that populations of predators and prey vary continuously and that birth rates and predation rates are numerically constant over time. These assumptions, while reasonable to a first approximation, do not hold in real world populations and, therefore, the solution to the differential equations is unlikely to yield accurate predictions. A stochastic model of predator/prey dynamics built in an agent-based language will not produce a formal equation as a result, but may produce better predictions of real world phenomena. Moreover, modeling also involves the deliberation over formal relations – it is the forms of expression of those relations that varies. In the case of models, the formal elements are the model rules. Conversely, since agent-based models can be refined at the level of causal rules, adjusting them is also more clearly an activity of trying to successively refine content-based rules until they yield satisfactory results (see e.g., Wilensky and Reisman, 1998; in press). Perhaps though, part of the discomfort with multi-agent models on the part of the formalist critics results from the symbolic form in which the model is captured. The history of mathematics and physics accustoms us to consider a physical phenomenon as captured by a model only when the symbolic form is an equation. When a phenomenon is captured by a set of rules as in a multi-agent model, it doesn't fit the paradigm and is rejected as incomplete. However, if we take the perspective that tools and forms of knowledge co-evolve, we might predict that the advent of widespread use of such models will lead to the acceptance of a new symbolic form for capturing system change, the form of agent-based rule-sets.

On the other hand, some educator critics of computer-based modeling have expressed concern that the activity of modeling on a computer removes children too much from the concrete world of real data (Tyack and Cuban, 1997; Stoll, 1999; Cordes and Miller, 2000). While it is undoubtedly true that children need to have varied and rich experiences away from the computer, the fear that computer modeling removes the child from concrete experience with phenomena is overstated. Indeed the presence of computer modeling environments invites us to reflect on the meaning of such terms as concrete experience (Wilensky, 1991). We come to see what we call 'concrete experience' as mediated by the tools and norms of our culture, and as such, subject to revision by a focused and enlightened cultural and/or pedagogic effort. This is particularly so with respect to scientific content domains in which categories of experience are in rapid flux and in which all experience is mediated by tools and instruments. In the GasLab case, it would be quite difficult to give learners "real-world" experience with the gas molecules. A real world GasLab experience would involve apparatuses for measuring energy and pressure that would be black boxes for the students using them. The range of possibilities for experiments that students could conduct would be much more severely restricted and would most probably be limited to the "received" experiments dictated by the curriculum and the tools it incorporates. Indeed, in a significant sense, the computer-based GasLab activity gives students a much more concrete understanding of the gas, seeing it as a macro-object that is emergent from the interactions of large numbers of micro-elements.

7. CONCLUDING REMARKS

In closing, a few additional remarks. As I have attempted to show in this paper, the use of model-based inquiry has the potential for significant impact on learning in this new century. We live in an increasingly complex and interconnected society. Simple models will no longer suffice to describe that complexity. Our science, our social policy, and the requirements of an engaged citizenry require an understanding of the dynamics of complex systems and the use of sophisticated modeling tools to display and analyze such systems. There is a need for the development of increasingly sophisticated tools that are designed for learning about the dynamics of such systems and a corresponding need for research on how learners, using these tools, begin to make sense of the behavior of dynamic systems. It is not enough to simply give learners modeling tools. Careful thought must be given to the conceptual issues that make it challenging for learners to adopt a systems dynamics perspective. The notion of levels of description, as in the micro- and macro-levels we have explored in this paper, is central to a systems dynamics perspective, yet is quite foreign to the school curriculum. Behavior such as negative and positive feedback, critical thresholds, dynamic equilibria are endemic to complex dynamic systems. It is important to help learners build intuitions and qualitative understandings of such behaviors. Side by side with modeling activity, there is a need for discussion, writing, and reflection activities that encourage students to reexamine some of the basic assumptions embedded in the science and mathematics curriculum: Assumptions that systems can be decomposed into isolated sub-systems, that causes add up linearly, that the causes have deterministic effects. In the Connected Learning projects we have seen, for example, how the 'deterministic mindset' (Resnick and Wilensky, 1993; Wilensky, 1997b) prevents students from understanding how stable properties of the world, such as Harry's Maxwell-Boltzmann distribution, can result from probabilistic underlying rules.

A pedagogy that incorporates the use of agent-based modeling tools for sustained inquiry has considerable promise to address such conceptual issues. By providing a substrate in which learners can embed their rules for individual elements and visualize global effects, this pedagogy invites learners to connect micro-level simulation with macro-level observation. By enabling them to control the behavior of thousands of agents in parallel, it invites them to see probabilism underlying stability and statistical properties as useful summaries of the underlying stochasm. By providing visual descriptions of phenomena that are too small or too large to visualize in the world, they invite a larger segment of society to make sense of such invisible phenomena. By providing a medium in which dynamic simulations can live and which responds to learner conjectures with meaningful feedback, it gives many more learners the experience of doing science and mathematics. A major challenge is to develop tools, pedagogy and policy that will bring this new form of literacy to the large majority of students and citizens.

ACKNOWLEDGMENTS

The preparation of this paper was supported by the National Science Foundation (Grants RED-9552950, REC-9632612, REC-9814682, REC-0126227). The ideas expressed here do not necessarily reflect the positions of the supporting agency. I would like to thank Seymour Papert for his overall support and inspiration and for his constructive criticism of this research in its early stages. Walter Stroup has been a close collaborator throughout the GasLab project. His ideas and contributions have been invaluable. Mitchel Resnick and David Chen gave extensive support in conducting the original GasLab research. I would also like to thank Josh Mitteldorf, Ed Hazzard, Christopher Smick and Seth Tisue for extensive discussions of the GasLab models and of the ideas in this paper. Wally Feurzeig, Nora Sabelli, Richard Noss, Ron Thornton, and Paul Horwitz made valuable suggestions to the project design. Seth Tisue, Brent Collins, Matt Goto, Sharona Levy, Ed Hazzard, Meridith Bruozas, Paul Deeds, Rob Froemke, Ken Reisman and Daniel Cozza contributed to the design and implementation of many of the more recent GasLab models. Bruce Sherin, Donna Woods and Dor Abrahamson made valuable comments and suggestions on a draft of this paper. Eric Betzold provided needed technical help with the figures.

NOTES

[1] The Connected Learning projects began at the MIT Media Lab in the late 1980s, moved to Tufts University in 1994 and to Northwestern University in 2000.

[2] Developed at the Center for Connected Learning and Computer-based Modeling, StarLogoT is freely downloadable from http://ccl.northwestern.edu/cm/.

[3] Developed at the Center for Connected Learning and Computer-based Modeling, NetLogo is freely downloadable from http://ccl.northwestern.edu/netlogo/.

[4] In the first five years of the Connected Probability project, students primarily used StarLogo (Connection Machine version) and StarLogoT. Since the late 1990s, students have primarily modeled in NetLogo.

[5] StarLogo (Resnick, 1994; Wilensky, 1995a), developed at the MIT Media Laboratory, was the first version of logo to be developed as a multi-agent language.

[6] Both StarLogoT and NetLogo permit users to change the dimensions of the patch grid, and thus the number of patches, though this is not as typical a use of the languages as is changing the number of turtles.

[7] Since NetLogo is Java-based, it can run on a large variety of computers. In most of our recent classroom work, students ran NetLogo on personal computers in our classroom or computer labs.

[8] As noted, Harry built this model in a version of StarLogo that ran on a connection machine parallel supercomputer. At the time Harry was building his model, StarLogo was not nearly as “user-friendly” as NetLogo is currently. This necessitated my working

together with Harry in constructing his model. Harry specified the behavior he wanted while I did most of the coding. As StarLogoT and NetLogo got more robust and easy to use, subsequent students were able to program the GasLab extensions themselves.

[9] Since the model was implemented in a general-purpose computer language, Harry could have calculated the exact collision points. The fact that he chose not to do so was partly due to the comparative ease of doing it stochastically. But, the primary reason for his choice lay in his inspiration from the original Maxwell (1860) papers, which had to take the stochastic approach for reasons of computational tractability.

[10] Harry's reasoning here appear to show one of the prototypical misconceptions about randomness. In previous work (Wilensky, 1993, 1995), I have shown many examples of this confusion in which random is interpreted to mean all possibilities are equally likely, rather than governed by some distribution. Harry's reasoning can also perhaps be thought of as an assimilation of individual collision events in the model in terms of some intuitive ("primitive") sense of 'balance' or 'compensation (see also diSessa, 1993).

[11] Although no law of energy conservation was explicitly programmed into the model, Harry knew that since the collisions were elastic, conservation of energy must hold true of the simulated gas.

[12] The fact that Harry was not content to get the correct result but felt the need also to understand what was wrong with his previous reasoning marks him as a sophisticated reasoner. A laudable goal for an inquiry-oriented curricular intervention would be to instill this habit of mind (Goldenberg, 1996) in learners.

[13] A source of confusion in many a physics classroom: Why do we need these two separate quantities, energy = mv^2 and momentum = mv. The algebraic formalism masks the big difference between the scalar energy and the vector momentum.

[14] As an example, suppose we have two particles each starting at speed 5. Then the sum of squares of their velocities is 50. After their collision, this sum of squares must stay constant. Say, one particle speeds up to 7 and the other slows down to 1 so that the sum of the squares of their speeds is still 50. But, originally the sum of their speeds was 10 and their average speed was 5, yet after the collision the sum of their speeds is 8 and their average speed has dropped to 4.

[15] Elsewhere we have described the condition of being very familiar with a formula but not understanding the rationale behind the formula as "epistemological anxiety" (Wilensky, 1997b). In this episode, we might say that Harry's epistemological anxiety about energy was relieved.

[16] To do this, they issued the simple NetLogo command: 'setxy 0 0'. Though the code for doing this is quite simple, this is not an experiment that can be replicated in the laboratory – a case of the model as an instantiation of ideal gas theory rather than of a real-world instantiation.

[17] This is one affordance of the GasLab environment that has been quite confusing to students and teachers in the Connected Chemistry Curriculum. They are used to seeing pressure as a flat line, not the ragged line they get in a GasLab model. We have taken this design constraint as a curricular focus, asking students to speculate why the graphs are different and asking them what interventions they could make to make it flatter. This focus has led to many productive discussions about what is a stable measure, what counts as a "fidelous" simulation and what simplifications are allowable when constructing a model.

[18] diSessa (1980) has argued that physics instruction should teach the concept of force in terms of momentum flow instead of as mass∗acceleration. This formulation fits better with student intuitions as well as expanding the applicability and importance of Newton's third law.

[19] Such as taking a sampling of fixed size small imaginary boxes within the physical box.

[20] In this regard, the students, though intrigued by the philosophical conundrum, chose the simpler pressure formulation absent any experimental consequence to changing it.

[21] He was ignoring the high speed limitation imposed by energy considerations.

[22] As one example of these reinvented thought experiments, they constructed a model of a divided box with a small opening in the divider in which a propeller is embedded. They measured the work done on the propeller by the particles hitting it and the propeller's consequent motion. A version of their model is downloadable from http://ccl.northwestern.edu/netlogo/models/GasLabSecondLaw.

[23] In the sense of understanding its reactions to typical human perturbations (see Wilensky, 1991; Noss and Hoyles, 1996b).

[24] When giving lectures on Harry's story in university physics departments, I have often seen physicists confused by the same issues as was Harry. For example, though they know that the Boltzmann distribution is asymmetric, they are often surprised to see more "blues" than "reds". They are not used to seeing the gas as a visual colored ensemble and do not readily convert their intuitions from the symbolic representations to the agent-based visualization.

[25] Typically, a hallmark of a good toolkit is that the desired application of the pieces, to a large degree, emerges from interacting with them. For instance, the dino "primitives" would probably constrain a construction of *some* large organism – perhaps a reptile – rather than a sonnet or an oil refinery.

[26] This is a simplification. Even so-called content neutral sets of primitives have affordances that make it easier to model some content domains than others. NetLogo, for example, makes it much easier to model phenomena that can be viewed as accumulations of large numbers of locally interacting elements. Processes that are composed of a small number of larger elements are less naturally modeled in NetLogo.

[27] Yet a second simplification is the dichotomous distinction between contentful model elements and neutral model elements. In selecting a set of primitive model elements, designer can choose to carve up the world in ways that more closely approximate the content domain and ways that are quite different than the content domain – there are a multiplicity of world slices and associated primitive sets.

[28] Similar points have been made, more generally, in relation to the design of external representations (see e.g., Norman, 1991; Zhang, 1993).

[29] In a recent book, Wolfram (2002) employs the forwards modeling approach by exploring the outcomes of possible rule sets of cellular automata.

[30] As one teacher comparing students' work with both STELLA and NetLogo models remarked: When students model with STELLA, a great deal of class time is spent on explaining the model, selling it to them as a valid description. When they do NetLogo modeling, the model is obvious, they do not have to be sold on it."

REFERENCES

Abrahamson, D. and Wilensky, U. (2003). The quest of the bell curve: A constructionist approach to learning statistics through designing computer-based probability experiments. *Proceedings of the Third Conference of the European Society for Research in Mathematics Education*. Bellaria, Italy, Feb. 28–March 3.

Bertalanffy, L. (1975). *General System Theory: Foundations, Development, Applications*. New York: George Braziller.

Bruckman, A. (1994). Programming for fun: MUDs as a context for collaborative learning. Paper presented at the *National Educational Computing Conference*. Boston, MA, June.

Bonabeau, E., Dorigo, M. and Théraulaz, G. (1999). *Swarm Intelligence: From Natural to Artificial Systems*. London: Oxford University Press.

Buldyrev, S., Erickson, M.J., Garik, P., Shore, L.S., Stanley, H.E., Taylor, E. F., Trunfio, P.A. and Hickman, P. (1994). Science research in the classroom. *The Physics Teacher* 32: 411–415.

Buchanan, M. (2002). *Nexus: Small Worlds and the Groundbreaking Science of Networks*. New York: W.W. Norton.

Camazine, S., Denoubourg, J., Franks, N., Sneyd, J., Theraulaz, G. and Bonabeau, E. (2001). *Self-Organization in Biological Systems*. Princeton, NJ: Princeton University Press.

Casti, J.L. (1994). *Complexification: Explaining a Paradoxical World Through the Science of Surprise*. New York: HarperCollins Publishers, Inc.

Centola, D. and Wilensky, U. (2000). Survival of the groupiest: Facilitating students' understanding of the multiple levels of fitness through multi-agent modeling – the EACH project. *Interjournal of Complex Systems*.

Centola, D., McKenzie, E. and Wilensky, U. (2000). A hands-on modeling approach to evolution: Learning about the evolution of cooperation and altruism through multi-agent modeling – the EACH project. *Proceedings of the Fourth Annual International Conference of the Learning Sciences*. Ann Arbor, MI, June 14–17.

Chen, D. and Stroup, W.M. (1993). General system theory: Toward a conceptual framework for science and technology education for all. *Journal of Science Education and Technology* 2(3): 447–459.

Collier, N. (2000). *RePast: An Extensible Framework for Agent Simulation*.

Cordes, C. and Miller, E. (2000). *Fool's Gold: A Critical Look at Computers in Childhood*. Collge Park, MD: Alliance for Childhood.

Cutnell, J. and Johnson, K. (1995). *Physics*. New York, Wiley & Sons.

Daston, L. (1987) Rational individuals versus laws of society: From probability to statistics. In L. Kruger, L. Daston and M. Heidelberger (Eds), *The Probabilistic Revolution*, Vol 1. Cambridge, MA: MIT Press.

Dawkins, R. (1976). *The Selfish Gene*. Oxford: Oxford University Press.

Dennett, D. (1995). *Darwin's Dangerous Idea: Evolution and the Meanings of Life*. New York: Simon and Schuster.

diSessa, A. (1980). Momentum flow as an alternative perspective in elementary mechanics. *American Journal of Physics* 48(5).

diSessa, A. (1986). Artificial worlds and real experience. *Instructional Science*: 207–227.

diSessa, A. (1993). Toward an epistemology of physics. *Cognition and Instruction* 10(2/3): 105–225.

diSessa, A. (2000). *Changing Minds: Computers, Learning, and Literacy*.

diSessa, A.A., Hoyles, C. and Noss, R. (Eds) (1995). *Computers and Exploratory Learning*. Berlin: Springer-Verlag (NATO ASI Series F, Volume 146).

Doerr, H. (1996). STELLA: Ten years later: A review of the literature. *International Journal of Computers for Mathematical Learning* 1(2).

Eisenberg, M. (1991). Programmable applications: Interpreter meets interface. *MIT AI Memo 1325*. Cambridge, MA: AI Lab, MIT.

Feurzeig, W. (1989). A visual programming environment for mathematics education. Paper presented at the *Fourth international Conference for Logo and Mathematics Education*. Jerusalem, Israel.

Forrester, J.W. (1968). *Principles of Systems*. Norwalk, CT: Productivity Press.

Gell-Mann, M. (1994). *The Quark and the Jaguar*. New York: W.H. Freeman.

Giancoli, D. (1984). *General Physics*. Englewood Cliffs, NJ: Prentice Hall.

Gigerenzer, G. (1987). Probabilistic thinking and the fight against subjectivity. In L. Kruger, L. Daston and M. Heidelberger (Eds), *The Probabilistic Revolution*, Vol 2. Cambridge, MA: MIT Press.

Ginsburg, H. and Opper, S. (1969). *Piaget's Theory of Intellectual Development*. Englewood Cliffs, NJ: Prentice-Hall.

Giodan, A. (1991). The importance of modeling in the teaching and popularization of science. *Trends in Science Education* 41(4).

Gleick, J. (1987). *Chaos*. New York: Viking Penguin.

Goldenberg, P. (1996). 'Habits of mind' as an organizer for the curriculum. *Journal of Education* 178(1): 13–34 (Boston University).

Hancock, C. (1995). The medium and the curriculum: Reflections on transparent tools and tacit mathematics. In A. diSessa, C. Hoyles and R. Noss (Eds), *Computers and Exploratory Learning* (pp. 221–240). Berlin: Springer.

Hofstadter, D. (1979). *Godel, Escher, Bach: An Eternal Golden Braid*. New York: Basic Books.

Holland, J.H. (1995). *Hidden Order: How Adaptation Builds Complexity*. New York: Addison-Wesley Publishing Company.

Holland, J.H. (1998). *Emergence: From chaos to order*. Reading, MA: Addison-Wesley Publishing Company, Inc.

Horwitz, P. (1989). ThinkerTools: Implications for science teaching. In J.D. Ellis (Ed.), *1988 AETS Yearbook: Information technology and Science Education* (pp. 59–71).

Horwitz, P., Neumann, E. and Schwartz, J. (1994). The Genscope project. *Connections*: 10–11.

Hoyles, C., Morgan, C. and Woodhouse, G. (Eds) (1999). *Rethinking the Mathematics Curriculum*. London: Falmer Press.

Jackson, S., Stratford, S., Krajcik, J. and Soloway, E. (1996). A learner-centered tool for students building models. *Communications of the ACM* 39(4): 48–49.

Johnson, S. (2001). *Emergence: The Connected Lives of Ants, Brains, Cities, and Software*. New York: Scribner.

Kafai, Y.B. (1998). Play and technology: Revised realities and potential perspectives. In D.P. Fromberg and D. Bergen (Eds), *Play from Birth to Twelve: Contexts, Perspectives, and Meanings* (pp. 93–99). New York: Garland Publishing.

Kauffman, S. (1995). *At Home in the Universe: The Search for the Laws of Self-Organization and Complexity*. Oxford: Oxford University Press.

Kay, A. (1991). Computers, networks and education. *Scientific American*: 138–148.

Kelly, K. (1994). *Out of Control*. Reading, MA: Addison Wesley.

Kennedy, J., Eberhart, R. and Shi, Y. (2001). *Swarm Intelligence*. San Francisco: Morgan Kaufmann.

Kruger, L. Daston, L. and Heidelberger, M. (Eds) (1987). *The Probabilistic Revolution*, Vol 1. Cambridge, MA: MIT Press.

Langton, C. and Burkhardt, G. (1997). *Swarm*. Santa Fe, NM: Santa Fe Institute.

Lotka, A.J. (1925). *Elements of Physical Biology*. New York: Dover Publications.

Mandinach, E.B. and Cline, H.F. (1994). *Classroom Dynamics: Implementing a Technology-Based Learning Environment*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Maxwell. C. (1860). *Illustrations of the Dynamical Theory of Gases*.

Mellar et al. (1994). *Learning With Artificial Worlds: Computer Based Modelling in the Curriculum*. Falmer Press.

Minar, N., Burkhardt, G., Langton, C. and Askenazi, M. (1997). *The Swarm Simulation System: A Toolkit for Building Multi-agent Simulations*. http://www.santafe.edu/projects/swarm/.

Minsky, M. (1987). *The Society of Mind*. New York: Simon & Schuster Inc.

Nemirovsky, R. (1994). On ways of symbolizing: The case of Laura and the Velocity Sign. *Journal of Mathematical Behavior* 14(4): 389–422.

Neuman, E., Feurzeig, W., Garik, P. and Horwitz, P. (1997). *OOTLS*. Paper presented at the European Logo Conference. Budapest, Hungary.

Norman, D.A. (1991). Cognitive artifacts. In J.M. Carroll (Ed.), *Designing Interaction: Psychology at the Human-Computer Interface* (pp. 17–38). New York: Cambridge University Press.

Noss, R., Healy, L. and Hoyles, C. (1997). The construction of mathematical meanings: Connecting the visual with the symbolic. *Educational Studies in Mathematics* 33(2): 203–233.

Noss, R. and Hoyles, C. (1996a). The Visibility of meanings: Modelling the mathematics of banking. *International Journal of Computers for Mathematical Learning* 1(1): 3–31.

Noss, R. and Hoyles, C. (1996b). *Windows on Mathematical Meanings: Learning Cultures and Computers*. Dordrecht, the Netherlands: Kluwer Academic Publishers.

Ogborn, J. (1984). A microcomputer dynamic modelling system. *Physics Education* 19(3).

Papert, S. (1980). *Mindstorms: Children, Computers, and Powerful Ideas*. New York: Basic Books.

Papert, S. (1991). Situating constructionism. In I. Harel and S. Papert (Eds), *Constructionism* (pp. 1–12). Norwood, NJ: Ablex Publishing Corp.

Papert, S. (1996). An exploration in the space of mathematics educations. *International Journal of Computers for Mathematical Learning* 1(1).

Papert, S. (2002). The Turtle's Long Slow Trip – Macro-educological Perspectives on Microworlds. *Journal of Educational Computing Research* 27(1–2), 2002.

Parker, M. (2001). What is Ascape and why should you care? *Journal of Artificial Societies and Social Simulation* 4(1).

Pea, R. (1985). Beyond amplification: Using the computer to reorganize mental functioning. *Educational Psychologist* 20(4): 167–182.

Piaget, J. (1929/1951). *The Child's Conception of the World*. London: Routledge and Kegan Paul.

Prigogine, I. and Stengers, I. (1984). *Order Out of Chaos: Man's New Dialogue with Nature*. New York: Bantam Books.

Repenning, A. (1993). *AgentSheets: A Tool for Building Domain-Oriented Dynamic, Visual Environments*. Ph.D. dissertation, Dept. of Computer Science, University of Colorado, Boulder.

Repenning, A. (1994). Programming substrates to create interactive learning environments. *Interactive learning environments* 4(1): 45–74.

Resnick, M. (1994). *Turtles, Termites and Traffic Jams. Explorations in Massively Parallel Microworlds*. Cambridge, MA: MIT Press.

Resnick, M. and Wilensky, U. (1998). Diving into complexity: Developing probabilistic decentralized thinking through role-playing activities. *Journal of the Learning Sciences* 7(2): 153–171.

Resnick, M. and Wilensky, U. (1995). New thinking for new sciences: Constructionist approaches for exploring complexity. *Presented at the Annual Conference of the American Educational Research Association*, San Francisco, CA.

Resnick, M. and Wilensky, U. (1993). Beyond the deterministic, centralized mindsets: New thinking for new sciences. *American Educational Research Association*. Atlanta, GA.

Richmond, B. and Peterson, S. (1990). *Stella II*. Hanover, NH: High Performance Systems, Inc.

Roberts, N. (1978). Teaching dynamic feedback systems thinking: An elementary view. *Management Science* 24(8): 836–843.

Roberts, N. (1981). Introducing computer simulation into the high schools: An applied mathematics curriculum. *Mathematics Teacher*: 647–652.

Roberts, N., Anderson, D., Deal, R., Garet, M. and Shaffer, W. (1983). *Introduction to Computer Simulations: A Systems Dynamics Modeling Approach*. Reading, MA: Addison Wesley.

Roberts, N. and Barclay, T. (1988). Teaching model building to high school students: Theory and reality. *Journal of Computers in Mathematics and Science Teaching* (Fall): 13–24.

Roetzheim, W. (1994). *Entering the Complexity Lab*. SAMS Publishing.

Sherin, B. (2001). A comparison of programming languages and algebraic notation as expressive languages for physics. *International Journal of Computers for Mathematical Learning* 6(1): 1–61.

Shore, L.S., Erickson, M.J., Garik, P., Hickman, P, Stanley, H.E., Taylor, E.F. and Trunfio, P. (1992). Learning fractals by ‘doing science’: Applying cognitive apprenticeship strategies to curriculum design and instruction. *Interactive Learning Environments* 2: 205–226.

Smith, D.C., Cypher, A. and Spohrer, J. (1994). Kidsim: Programming agents without a programming language. *Communications of the ACM* 37(7), 55–67.

Starr, P. (1994). Seductions of Sim. *The American Prospect* (17).

Stieff, M. and Wilensky, U. (in press). The Connected chemistry modeling environment: Incorporating interactive simulations into the chemistry classroom. *Journal of Science Education and Technology*.

Stieff, M. and Wilensky, U. (2002). ChemLogo: A novel computer-based modeling environment for teaching and learning chemistry. *Proceedings of the Fifth Biannual International Conference of the Learning Sciences*. Seattle, WA, October.

Stoll, C. (1999). *High-Tech Heretic: Why Computers Don’t Belong in the Classroom and Other Reflections by a Computer Contrarian*. New York: Doubleday.

Stroup, W. (2002). Understanding qualitative calculus: A structural synthesis of learning research. *International Journal of Computers for Mathematical Learning* 7(2): 167–215.

Thornton, R. and Sokoloff, D. (1990). Learning motion concepts using real-time microcomputer-based laboratory tools. *Am. J. of Physics* 58: 9.

Tipler, P. (1992). *Elementary Modern Physics*. New York: Worth Publishers.

Turkle, S. (1995). *Life on the Screen: Identity in the Age of the Internet*. New York: Simon and Schuster,

Tversky, A. and Kahneman, D. (1974). Judgment under uncertainty: Heuristics and biases. *Science* 185: 1124–1131.

Tyack, D. and Cuban, L. (1997). *Tinkering Toward Utopia: A Century of Public School Reform*. Cambridge, MA: Harvard University Press.

Tymoczko, T. (1979). The four color problem and its philosophical significance. *The Journal of Philosophy* (February) LXXVI(2).

Ventana Systems (2002). *Vensim*. Harvard, MA: Ventana Systems, Inc.

Waldrop, M. (1992). *Complexity: The Emerging Order at the Edge of Order and Chaos*. New York: Simon & Schuster.

White, B. and Frederiksen, J. (1998). Inquiry, modeling, and metacognition: Making science accessible to all students. *Cognition and Instruction* 16(1): 3–118.

Wilensky, U. (2001). Modeling nature's emergent patterns with multi-agent languages. *Proceedings of EuroLogo 2001*. Linz, Austria.

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

Wilensky, U. (1997a). *StarLogoT*. Evanston, IL: Center for Connected Learning and Computer-Based Modeling, Northwestern University. http://ccl.northwestern.edu/cm/.

Wilensky, U. (1997b). What is normal anyway? Therapy for epistemological anxiety. *Educational Studies in Mathematics* 33(2): 171–202. Special Edition on *Computational Environments in Mathematics Education*, R. Noss (Ed.).

Wilensky, U. (1996). Modeling rugby: Kick first, generalize later? *International Journal of Computers for Mathematical Learning* 1(1).

Wilensky, U. (1995a). Paradox, programming and learning probability: A case study in a connected mathematics framework. *Journal of Mathematical Behavior* 14(2).

Wilensky, U. (1995b). Learning probability through building computational models. *Proceedings of the Nineteenth International Conference on the Psychology of Mathematics Education*. Recife, Brazil, July 1995.

Wilensky, U. (1993). *Connected Mathematics: Building Concrete Relationships with Mathematical Knowledge*. Doctoral dissertation, Cambridge, MA: Media Laboratory, MIT.

Wilensky, U. (1991). Abstract meditations on the concrete and concrete implications for mathematics education. In I. Harel and S. Papert (Eds), *Constructionism*. Norwood NJ: Ablex Publishing Corp.

Wilensky, U. and Reisman, K. (in press). Thinking like a wolf, a sheep or a firefly: Learning biology through constructing and testing. Computational theories – an embodied modeling approach. *Cognition and Instruction*.

Wilensky, U. and Reisman, K. (1998). Learning biology through constructing and testing computational theories – an embodied modeling approach. In Y. Bar-Yam (Ed.), *Proceedings of the Second International Conference on Complex Systems*. Nashua, NH: New England Complex Systems Institute.

Wilensky, U. and Resnick, M. (1999). Thinking in Levels: A Dynamic Systems Perspective to Making Sense of the World. *Journal of Science Education & Technology* 8(1): 3–18.

Wilensky, U. and Stroup, W. (1999). Learning through participatory simulations: Network-based design for systems learning in classrooms. *American Educational Research Association*. Montreal, Canada.

Wilensky, U and Stroup, W. (2000). Networked Gridlock: Students Enacting Complex Dynamic Phenomena with the HubNet Architecture. *Proceedings of the Fourth Annual International Conference of the Learning Sciences*. Ann Arbor, MI, June 14–17, 2000.

Wilensky, U. and Stroup, W. (2002). Participatory Simulations: Envisioning the networked classroom as a way to support systems learning for all. Presented at the annual meeting of the *American Educational Research Association*. New Orleans, LA.

Wilensky, U., Hazzard, E and Froemke, R. (1999). An extensible modeling toolkit for exploring statistical mechanics. *Proceedings of the Seventh European Logo Conference* – EUROLOGO'99. Sofia, Bulgaria.

Wright, W. (1992a). *SimCity*. Orinda, CA: Maxis.
Wright, W. (1992b). *SimEarth*. Orinda, CA: Maxis.
Zhang, J. (1993). External representation: An issue for cognition. *Behavioral and Brain Sciences* 16(4): 774–775.

Center for Connected Learning and Computer-Based Modeling
Northwestern University
http://ccl.northwestern.edu