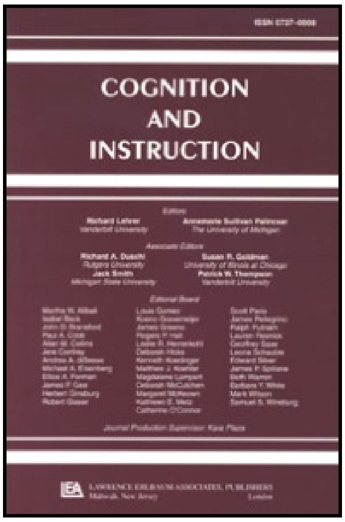

Cognition and Instruction
Publication details, including instructions for authors and subscription information:
http://www.informaworld.com/smpp/title~content=t775648096

Thinking Like a Wolf, a Sheep, or a Firefly: Learning Biology Through Constructing and Testing Computational Theories—An Embodied Modeling Approach

Uri Wilensky [a]; Kenneth Reisman [b]
[a] Departments of Learning Sciences and Computer Sciences, Northwestern University. [b] Stanford University.

To cite this Article Wilensky, Uri and Reisman, Kenneth(2006) 'Thinking Like a Wolf, a Sheep, or a Firefly: Learning Biology Through Constructing and Testing Computational Theories—An Embodied Modeling Approach', Cognition and Instruction, 24: 2, 171 — 209
To link to this Article: DOI: 10.1207/s1532690xci2402_1
URL: http://dx.doi.org/10.1207/s1532690xci2402_1

COGNITION AND INSTRUCTION, *24*(2), 171–209

Thinking Like a Wolf, a Sheep, or a Firefly: Learning Biology Through Constructing and Testing Computational Theories— An Embodied Modeling Approach

Uri Wilensky
Center for Connected Learning and Computer-Based Modeling
Departments of Learning Sciences and Computer Sciences
Northwestern University

Kenneth Reisman
Stanford University

Biological phenomena can be investigated at multiple levels, from the molecular to the cellular to the organismic to the ecological. In typical biology instruction, these levels have been segregated. Yet, it is by examining the connections between such levels that many phenomena in biology, and complex systems in general, are best explained. We describe a computation-based approach that enables students to investigate the connections between different biological levels. Using agent-based, embodied modeling tools, students model the microrules underlying a biological phenomenon and observe the resultant aggregate dynamics. We describe 2 cases in which this approach was used. In both cases, students framed hypotheses, constructed multiagent models that incorporate these hypotheses, and tested these by running their models and observing the outcomes. Contrasting these cases against traditionally used, classical equation-based approaches, we argue that the embodied modeling approach connects more directly to students' experience, enables extended investigations as well as deeper understanding, and enables "advanced" topics to be productively introduced into the high school curriculum.

Correspondence should be addressed to Uri Wilensky, Annenberg 311, 2120 Campus Drive, Northwestern University, Evanston, IL 60208. E-mail: uri@northwestern.edu

> When I observed phenomena in the laboratory that I did not understand, I would also ask questions as if interrogating myself: "Why would I do that if I were a virus or a cancer cell, or the immune system" Before long, this internal dialogue became second nature to me; I found that my mind worked this way all the time. (Salk, 1983, p. 7)

There is a sharp contrast between the picture of the field of biology as studied in school settings and the picture that emerges from the practice of current biology research. Although the two pictures are linked by similar content and the objects of study are recognizably the same, the *processes* involved in the two activities are quite different.

In school settings, typical instruction emphasizes the memorization of classification schemas and established theories. In middle school, classification may take the form of learning the names of the bones of the body, the names and shapes of different plant leaves or the phyla in the animal kingdom. In high school and early undergraduate studies, the content broadens to include unseen phenomena such as parts of the cell or types of protozoa, but the processes of memorizing classifications remains essentially the same. Similarly, students study biological explanation by absorbing established theories about the process of photosynthesis, the Krebs cycle, or the succession of evolutionary ancestors. Even in cases in which the theories are not yet established, such as the extinction of the dinosaurs, the alternative theories are presented as competing stories to be memorized. Even when students are exposed to research techniques in laboratory work, the emphasis is on following a prescribed procedure rather than reasoning from the evidence gathered in the procedure.

This picture contrasts sharply with the picture that emerges from the recent biology research literature (e.g., Keeling & Gilligan, 2000; Marion, Renshaw, & Gibson, 2000). In this picture, the participants are active theorizers. They gather new evidence and devise methods to test their theories. Instead of accepting classifications as given, they see these classifications as provisional theories that are constantly reassessed and reconstructed in light of the dialogue between theory and evidence. Participants also reason both forwards, by constructing theories that are consistent with the known evidence, and backwards, by deducing consequences of theories and searching for confirming–disconfirming evidence. In constructing or assessing an account of a biological phenomenon, they focus on the plausibility of the *mechanism* proposed—can it achieve the task assigned it in a biologically feasible manner? This assessment of the mechanism often involves reasoning across a range of levels; thus, participants ask, is the mechanism constrained by the structure at the molecular, the cellular, the organismic or the ecological level?

The contrast between the processes in which these two communities are engaged leads biology students to form a misleading picture of the biological research enterprise. Students form beliefs that biology is a discipline in which observation and classification dominate and reasoning about theories is rare. Furthermore, they believe that learning biology consists of absorbing the theories of experts and that constructing and testing their own theories is out of reach.[1]

In this article, we present an approach that attempts to narrow the gap between school biology and research biology. The approach centers on the use of innovative computer modeling tools that enable students[2] to learn biology through processes of constructing and testing theories.

In recent years, a number of educational research projects (Feurzeig & Roberts, 1999; Gobert et al., 2004; Jackson, Stratford, Krajcik, & Soloway, 1996; Jacobson & Kozma, 2000; Jungck & Calley, 1985; Lehrer & Schauble, 2000; Ogborn, 1999; Roberts & Barclay, 1988; White & Horwitz , 1988) have used computer-modeling tools in science instruction. The approach taken herein differs from these approaches in its use of *agent-based modeling languages* (also known as *object-based parallel modeling languages*) that enable students to model biological elements at the level of the individual (e.g., individual wolf–sheep) as opposed to aggregate (differential equation-based) modeling languages that model at the level of the population (wolf–sheep populations). This technical advance in modeling languages enables students to use their knowledge of the behavior of individual organisms (or molecules, cells, genes, and so on) in the construction of theories about the behavior of populations of organisms. Furthermore, the ability to model individual behavior enables students to use their personal experience with sensing and locomoting in the world as initial elements in their models of other organisms. In this way, the well-known tendency of children to explain biological behavior through personification (see Carey, 1986; Hatano & Inagaki, 1987), instead of being seen as a misconception or a limitation to be overcome, becomes a building block toward the construction and refinement of plausible biological explanations.[3]

In previous work, we and other agent-based modeling designers (Repenning, 1994; Resnick, 1994; D. C. Smith, Cypher, & Spohrer, 1994; Wilensky, 1995, 2001; Wilensky & Resnick, 1999) have described the approach in a broad interdisciplinary context. In this article, we explore the use of this approach, specifically, in biology instruction.

[1] For a similar and much fuller account of the gap between school and research physics, see Hammer (1994).

[2] Elsewhere (Tisue & Wilensky, 2004; Wilensky, 2001) we have argued that the availability and ease of use of agent-based modeling environments have also enabled scientists to conduct experimental research that was difficult or impossible to do with traditional methods.

[3] For a detailed discussion on misconceptions reconceived, see J. P. Smith, diSessa, and Roschelle (1994).

MATHEMATICAL BIOLOGY AND COMPUTER-BASED MODELING— IN THE FIELD AND IN THE CLASSROOM

The gap between school biology and research biology can be partially explained by a lag in the transfer of newer biological methods to the school setting. Indeed, at all levels from the molecular to the ecological, the science of biology has undergone an important shift over the last century. As biologists have increasingly availed themselves of the language of dynamic systems to model natural phenomena, biology—once an entirely qualitative discipline—has become more quantitative.[4]

Mathematical models have added precision to biological theories, have increased their predictive power, and have been important sources of explanations and hypotheses. The generation and refinement of such models has become a pervasive element of modern biological inquiry. Yet, despite this virtual revolution in biology practice, the high school and undergraduate biology curriculum have scarcely noticed. For most secondary and postsecondary biology students, the study of biology remains primarily an exercise in memorization. Because of the formidable mathematical prerequisites that quantitative models of biological change have traditionally imposed, students below the advanced undergraduate level are given little or no exposure either to dynamic models or to the process of modeling biological change. The computational approach presented here enables us to give students this exposure while sidestepping the traditional mathematical roadblocks.

We begin, in the following section, by describing our "embodied" approach to biological modeling and the agent-based modeling languages, StarLogoT (Wilensky, 1997a) and NetLogo (Wilensky, 1999b), in which the models are constructed. In the next section, we illustrate this approach and contrast it with classical modeling techniques by developing both embodied and classical models of predator–prey population fluctuations. We follow a high school student, Talia, in her efforts to create embodied models of wolf–sheep predation. In the next section, we follow another student, Paul, as he develops a computational model of synchronously flashing fireflies (these species of fireflies are prevalent in the Far East, especially Thailand). In contrast to the topic of predator–prey population dynamics, the firefly flash synchronization problem does not easily admit classical approaches and is, thus, unfamiliar to students. We use this example to frame a discussion of the student modeling process and the relation of this process to modeling within science. Finally, in our concluding remarks we respond to criticism of our approach and summarize the major points of the article.

[4] For an illuminating discussion of this transformation of the biological field, see Allen (1975).

RESEARCH SETTINGS

The student modelers described in this article were participants in the Connected Mathematics (Wilensky, 1993, 1995) and, principally, the Making Sense of Complex Phenomena (MSCP; Wilensky, 1997b, 2001) projects in which students learn about complex systems through construction of agent-based models of these systems. The goals of the MSCP project are to construct computational toolkits that enable students to model complex systems, to study students engaged in using these toolkits, and to make sense of their behavioral dynamics. Research has documented the difficulties people have in making sense of emergent phenomena, global patterns that arise from distributed interactions, central to the study of complex systems. We have labeled the constellation of difficulties in understanding emergent phenomena and constructing distributed explanations of such phenomena the *deterministic–centralized* mindset (Resnick, 1996; Resnick & Wilensky, 1993; Wilensky & Resnick, 1995, 1999). In the MSCP project we have worked with a wide variety of students, ranging from middle school students to graduate student researchers as well as both preservice and in-service teachers, on moving beyond this mindset to a richer understanding of the dynamics of complex systems. The primary research sites are two urban Boston high schools. Students from these schools participated in the project as part of their classroom work. Undergraduates and preservice teachers participated in the context of teacher education courses at Tufts University. Some students participated through informal contexts, pursuing modeling investigations in afterschool settings or at the laboratory, housed at the project site, the Center for Connected Learning and Computer-Based Modeling (then at Tufts University, now moved to Northwestern University). In the classroom context, students, typically, were involved in an extended classroom modeling project led by the classroom teacher and assisted by project researchers. The role of the researchers was to document student work through videotaping and field notes and to support students and teachers in the use of project materials and modeling languages. Examples of such support included bringing in books and Web sites that might be useful to the modelers and disseminating interesting cases as potential sources of models. Project researchers also engaged students in structured activities (including participatory simulations not involving the computer; Resnick & Wilensky, 1998) that would foster reflection on the concept of emergence. They also provided support to students and teachers on the syntax of the modeling language. The computational models described in this article were built in an agent-based (also known as multiagent or object-based parallel) modeling language called StarLogoT (Wilensky, 1997a).[5] Re-

[5] StarLogoT is one of several variants of the StarLogo modeling language (Resnick, 1994, 1996; Resnick & Wilensky, 1993; Wilensky & Resnick, 1999)—an extended version of MacStarLogo. StarLogoT and the models described in this article can be downloaded from ccl.northwestern.edu/cm/. Updated and more complete NetLogo versions of these models can be downloaded from ccl.northwestern.edu/netlogo/models.

cently, Uri Wilensky developed another agent-based modeling language, NetLogo (Wilensky, 1999b), that is a (more sophisticated) successor to StarLogoT. The models described herein have been converted to NetLogo.[6] As NetLogo is the more recent and more powerful environment, for the remainder of this article we refer to NetLogo and not to StarLogoT. In the next section, we describe the workings of the NetLogo language and its advantages for modeling biological phenomena.

THE NETLOGO MODELING LANGUAGE

NetLogo derives from, and has contributed to, recent work in the field of complex systems. This field studies the dynamics of systems that are constituted by many interacting elements. Taken as a whole, the behavior of these systems can be extremely complex and difficult to predict, though their individual elements may be quite simple. Examples can be found in many fields, from physics and chemistry to economics and political science. Biology has been a particularly fertile domain of complex-systems-oriented research (Kauffman, 1995; Langton, 1994). Indeed, though much of the early work in complex systems theory originated in physics (and to this day, the primary echo of this research in classrooms is in physics classes), in our view, the paradigmatic area for studying complexity lies in the study of complex biological systems. Though the brain, the immune system, and the behavior of organisms such as ants or bees are all oft-cited examples, in fact, nearly all of biology can be considered from a complex systems perspective. Genetic and cellular processes can be viewed as the complex outcomes from molecular interactions; organisms and their organs can be viewed as the complex outcomes from cellular- and genetic-level interactions; and ecological systems can be viewed as the complex outcomes of interactions between individual organisms. Of course, there is causality in the other direction as well; organism behavior can affect cellular and genetic level activity, and ecological circumstances can affect the behavior of individuals. Indeed, one reason complex systems can be so difficult to study is that aggregate-level structures can have feedback effects on the behavior of the elements of which they are composed.

NetLogo is a general-purpose (domain-independent) modeling language and integrated environment. It works by providing the modeler with a framework to represent the basic elements—the smallest constituents—of a system and then provides a way to simulate the interactions between these elements. With NetLogo, students write rules for hundreds or thousands of these basic elements, specifying how they should behave and interact with one another. These individual elements are referred to as *turtles*. (NetLogo owes the turtle object to the Logo computer lan-

[6] The application NetLogo and all the models here can be freely downloaded from the NetLogo site at ccl.northwestern.edu/netlogo/.

guage.[7]) Turtles are situated on a two-dimensional grid on which they can move around. Each cell on the grid is called a "patch," and patches may also execute instructions and interact with turtles and other patches. Some typical commands for a turtle move in a given direction, change color, set a variable according to some value, "hatch" new turtles, or look at the properties (variables) of other turtles. Turtles can also generate random values, so that they can, for example, execute a sequence of commands with a fixed probability. Patches can execute similar commands, though they cannot change location. The wide range of commands executable by turtles and patches makes it possible to use them to represent many different systems. For example, turtles can be made to represent molecules, cells, or individual organisms, whereas patches can represent the medium (whatever it may be) in which they interact.[8]

Dynamic modeling tools, such as NetLogo, are used to represent changes in the states of systems over time. In NetLogo, time is represented as a discrete sequence of "clock-ticks." At each clock-tick, each turtle and patch is called on to execute the rules that have been written for it. Students need not write separate rules for each turtle (or patch)—the power of NetLogo comes from the fact that all turtles can execute the very same set of rules at each clock-tick. If all turtles are executing the same rules, will their collective behavior not be repetitive and uninteresting? To see why this is not the case, it is important to take note of the fact that even though two turtles might be following the same rules, their behavior could be markedly different. This possibility exists because the two turtles may have quite different internal properties and may be situated in dissimilar environments. For example, the turtles may be following the rule "If you smell food ahead, move forward a distance equal to your body length. Otherwise, turn around." If one turtle is in the vicinity of food, it will move forward, the other turtle, far from the food, will turn around. Even if they are both in the vicinity of food, and even in the exact same location, if they have different body measurements, they will move to different locations. It is this diversity in internal states and in surrounding environs that enables the collective turtle behaviors to admit a surprising degree of variance.

The modeling approach we describe—instantiating the individual elements of a system and simulating their interactions—is not unique to NetLogo. Such models have been used across a wide variety of domains and have been referred to by many different labels, such as *object-based parallel models* (Wilensky, 1995, 1996) *agent-based models* (Beer, 1990; Epstein & Axtell, 1996; Maes, 1990; Repenning, 1993), *multi-agent models* (Jacobson et al., 1998; Wilensky, 2001), *individual-based models* (Huston, DeAngelis, & Post, 1988; Judson,

[7] In Logo (Feurzeig, 1984; Papert, 1980), children type commands to a graphical turtle, and the turtle draws geometric figures on the computer screen.

[8] Both turtles and patches, as core functional elements of the system, are also referred to as "agents."

1994), and *particle simulations* (Buneman, Barnes, Green, & Nielsen, 1980). These "new wave" modeling approaches have transformed biology research practice[9] and enabled researchers to model increasingly complex multileveled biological systems (Forrest, 1989; Keen & Spain, 1992; Langton, 1994; Taylor, Jefferson, Turner, & Goldman, 1989). For the remainder of this article, we use the term *embodied modeling* to refer to this general approach. Although the other terms described earlier, which we have used in the past, are perhaps a more accurate description of the technical workings of NetLogo, the *embodied modeling* label more closely matches the experience of a biology modeler who is actively engaged in understanding and embodying the behavior of individual biological elements.

In the following two sections of the article, we illustrate the embodied modeling approach in biology with two extended examples of modeling biological phenomena. We intend these examples to illustrate how such an approach can both (a) facilitate the creation and verification of predictive multilevel models in biology and (b) enable biology students to create more powerful explanations of and deepen their understanding of biological phenomena.

MODELING PREDATOR–PREY POPULATION DYNAMICS[10]

The dynamics of interacting populations of predators and their prey have long been a topic of interest in population biology. Comparisons of a number of case studies have revealed similar dynamics between such populations, regardless of the specific species under study and the details of their interactions (Elton, 1966). Notably, when the sizes of the predator and prey populations are compared over many generations, we tend to find regular oscillations in these sizes that are out of phase; where one increases, the other tends to decline, and vice versa (see Figure 1). Numerous mathematical models have been proposed to explain these oscillations. In this section, we examine several NetLogo models that are at considerable variance from classical versions. Along with providing a firsthand glimpse of our approach to modeling systems, the example also allows us to contrast the different perspectives promoted by embodied versus classical tools. We begin with a look at a well-known classical model.

[9] In biology, the term most often used for this kind of modeling is *individual-based* modeling.

[10] The predator–prey model and numerous other models (collectively known as *connected models*; Wilensky, 1998) can be downloaded from http://ccl.northwestern.edu/netlogo/models.

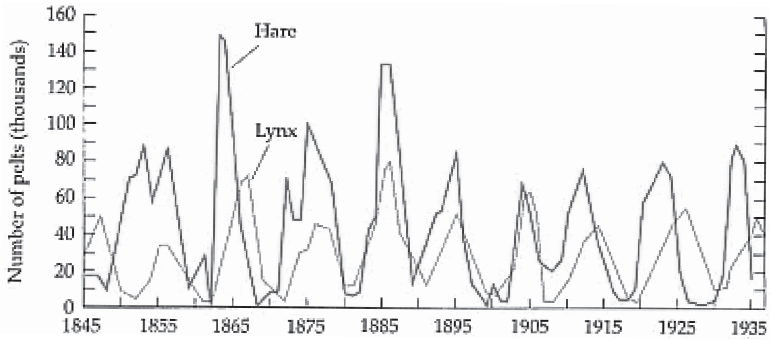

FIGURE 1 Fluctuations of the sizes of predatory lynx and prey hare populations in Northern Canada from 1845 to 1935. From *Life: The Science of Biology* (3rd ed., Figure 46.7B, p. 1060), by W. Purves, G. Orian, and H. Heller, 1992, Sunderland, MA: Sinauer. Copyright 1992 by Sinauer Associates. Reprinted with permission.

The Classical Approach

For many years, models of predation were based on the Lotka–Volterra (Lotka, 1925; Volterra, 1926) model.[11] Alfred Lotka and Vito Volterra (working independent of each other) were among the first to carry over to biology differential equation models, previously used principally in physics and chemistry. The Lotka–Volterra model of predation works by specifying interactions between the predator and prey populations framed as a set of coupled differential equations. Each such equation describes the rate at which a given variable (e.g., the density of the prey population) changes over time. Here we present the Lotka–Volterra predation equations, which describe changes in the densities of the prey population (N_1) and the predator population (N_2). Keep in mind that population size and population density are proportional to one another.

$$dN_1/dt = b_1N_1 - \mathrm{k}_1N_1N_2 \qquad (1)$$

$$dN_2/dt = k_2N_1N_2 - d_2N_2 \qquad (2)$$

In these equations b_1 is the birth rate of the prey, d_2 is the death rate of the predators, and k_1 and k_2 are constants.

Let us briefly analyze Equation 1. There are two terms in the equation. In the first term (b_1N_1), the prey birth rate is multiplied by the density of the prey popula-

[11] For some recent predation models, as well as some other classical biological models see Murray (1989).

tion, yielding the increase in density due to new prey births. In the second term ($k_1 N_1 N_2$), the frequency of interaction between the prey and the predator populations is determined, yielding the decrease in prey density due to consumption of prey by predators. The rate of change in the density of the prey population is thus computed by subtracting the total effect of prey deaths from the total effect of prey births. Equation 2 can be analyzed along similar lines, although in this equation predator births are dependent on the frequency of predator–prey interactions, whereas predator deaths are not—a reversal from Equation 1.

It is important to notice that Equation 1, which describes the prey population, contains N_2, the variable describing the density of the predator population, and vice-versa. The equations thus specify how the density (and so the size) of each population depends on the density of the other. Specifically, increases in the prey population will cause the predator birth rate to rise, and increases in the predator population cause the prey death rate to rise. A typical plot produced from these equations is shown in Figure 2. Indeed, we see here the characteristic cyclical fluctuations between the predator and prey populations.

We need not go in further depth about these equations. The point to notice for now is that the classical approach describes the cyclical fluctuations between predator and prey populations by specifying relations between population-level properties, such as birth rate, frequency of interaction, and overall density.

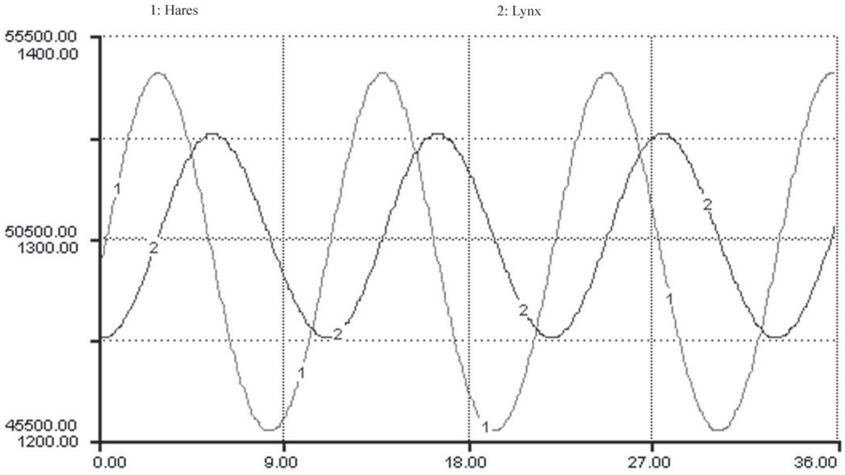

FIGURE 2 Results of the Lotka–Volterra predation model for lynx and hares with respective initial populations of 1,250 and 50,000.

The Embodied Approach

Using embodied tools, such as NetLogo, we approach this problem from a different angle. Rather than describe relations between properties of populations, we are concerned primarily with specifying the behavior of individuals. The relevant question here is, what kinds of actions must an individual predator or individual prey follow so that populations of such individuals will exhibit the characteristic oscillations? Another way to think about the actions of individuals—the method behind NetLogo modeling—is to consider the rules that each organism might follow in order for the given population-level patterns to result.

There are a number of paths that a modeler might take toward finding such a set of rules (indeed, there are often a number of equally effective solutions). It may seem to readers that one would need to be highly familiar with the phenomenon being modeled and with current theories in order to make meaningful progress, but our experience indicates otherwise. In the MSCP project, we have found that with only a small amount of background knowledge, students are often able to develop solid explanatory models of various phenomena. We generally encourage modelers to try to make sense of a problem on their own before seeking external resources, and often they are quite surprised at how far they are able to get. Rather than quickly reaching for the "facts," students undertake something akin to a scientific inquiry, and they generally learn much more than if someone had simply given them the solution. Of course, the body of existing research is quite important to the development of a model, and NetLogo modelers will often go back and forth between developing new hypotheses and researching existing solutions. To help convey a sense of this process, we describe the development of a NetLogo predation model from the standpoint of two students, Talia and Benny. These students were fairly typical of the secondary student modelers who participated in the MSCP project.

Finding Rules for Wolves—An Initial Model

Talia's task was to formulate a plausible set of rules for a predator and a prey. Recall that the characteristic properties of predator–prey population dynamics have been observed to be strongly similar across many species and many different conditions. Rather then being specific, then, these rules had to point to general behaviors that all such species perform in one way or another. In her first attempt, she described a predator (say, a wolf) as moving about in the NetLogo world and looking for prey (say, sheep). As a real wolf needs energy to live, she decided that each step in the world should cost the model wolf energy. Running out of energy will cause the wolf to die, and the only way to gain energy is by eating sheep. In this way, Talia was, also, as in the aforementioned Lotka–Volterra model, describing a dependency between predators and prey: Wolves are likely to persist when sheep are abundant (because they are unlikely to run out of food–energy), and sheep are expected to die when wolves are abundant (because they will eventually be eaten). Here, then, is a simple rule set for a wolf based on Talia's initial description:[12]

Rule Set W1: wolf
at each clock-tick:

1. move randomly to an adjacent patch and decrease energy by E_1
2. if on the same patch as one or more sheep, then eat a sheep and increase energy by E_2
3. if energy < 0 then die
4. with probability R_1 reproduce

Talia decided on a simpler rule set for the sheep. They only move about and reproduce, though they risk being eaten by the wolves:

Rule Set S1: sheep
at each clock-tick:

1. move randomly to an adjacent patch
2. with a probability of R_2, reproduce

Notice that the mechanism for reproduction in Talia's model is blind probability; any wolf or sheep may reproduce at a given clock-tick if the numbers come up right. This may seem like she was cheating, for surely this is an unrealistic way to portray behavior at the individual level. Talia had a firm justification for this though. She reasoned that there are many different ways in which various organisms reproduce, and yet, similar dynamics tend to arise in populations regardless of the specific reproductive mechanisms. To keep the model as general as possible, she adopted a probabilistic rule that effectively says "reproduce every now and then." This rule allowed her to achieve the desired behavior without being specific about mechanisms.

Of course, mechanisms are important to embodied models, and they generally are specified. The rules governing death in this model, for example, are more specific than those for reproduction: Prey die specifically when they are eaten by predators, and predators die specifically by running out of food. Wherever the particular mechanism is relevant to the model, it should be included; otherwise, details in the model can be minimized using probabilistic rules.

To be sure, there are many simplifications made by Talia's model that are questionable. A quick list includes only a single factor limiting the growth of predators (starvation), only a single factor limiting the growth of prey (they are eaten), random movement, no limit on number of organisms on a single patch, only two di-

[12] The rule sets are stated in summary form here. For the actual NetLogo code, please visit http://ccl.northwestern.edu/netlogo/models/predation/.

mensions, and so forth. It is certainly possible, even likely, that these are not just simplifications, but oversimplifications. There is no quick way to determine where such abstractions are valid and where they are mistaken. This uncertainty, though, is an integral part of the process of modeling—not only with embodied models, but with any scientific modeling process. The modeler must carefully consider which kinds of simplifications are plausible, and, even then, it is often only repeated testing of the model and revision of the assumptions that may ultimately lead to a valid model.

Once Talia had completed the coding of her model in the NetLogo language, she selected values for each of the parameters in the model (i.e., the E1, E2, R1, R2 parameters of Rule Set 1 as well as the initial number of wolves and sheep, and the length and width of the patch world). The values of model parameters, initially set by intuition, will often have a significant effect on the outcome of a NetLogo simulation. The modeler may induce what kind of effects, if any, each parameter has on the outcome by repeatedly altering these parameters and observing the result. Because the relation between the various parameters of a model can be nonlinear, this can be, not surprisingly, a difficult task.

After Talia ran her model several times under various parameter configurations, she noted that one of two general outcomes would always result. Most often, the populations oscillated until all the sheep were eaten, whereupon the wolves died from starvation (see Figure 3, right panel). Sometimes—usually under low density parameters (e.g., small population size or narrow screen width)—there were oscillations until the number of sheep dipped too low and the wolves all died off, at which point the sheep population increased at an exponential rate (see Figure 3, left panel). Thus, Talia's rule set successfully produced population oscillations, but this pattern was consistently transient and unstable. These results were clearly not in line with the sustained population oscillations observed in nature and those of

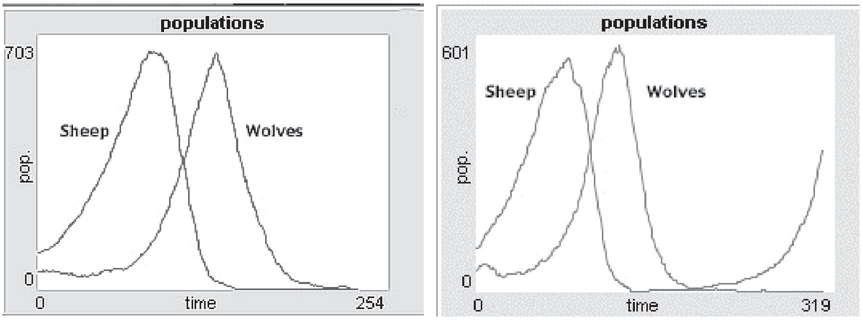

FIGURE 3 Two different outcomes from Rule Sets W1 + S1. In the left panel, both populations (pop.) go extinct. In the right panel, the wolves go extinct, and therefore the sheep "inherit the earth."

the Lotka–Volterra model. The next logical step in the modeling process was for her to revise her thinking.

Revising the Model

Talia was initially disturbed to find that her model did not meet her expectations. Still, she was determined to create a version that exhibited a stable relation between the wolf and sheep populations. That is, one where the two populations would continue to coexist, despite ongoing fluctuations in size. To generate such a version, she put great effort into understanding the behavior of her existing model. She asked herself "Why did the populations crash? What factors might be missing or misrepresented?" From analyzing plots such as those shown in Figure 3, she was led to several observations concerning the stability of her model. First, the peaks of the wolves follow peaks of the sheep. Second, the higher the peaks of the sheep, the higher the peaks of the wolves. Third, the higher the peak, the deeper the following crash. Instability in the model appeared to manifest itself in the ever-increasing amplitude of the population oscillations. The peaks get higher and the crashes get deeper over time, until zero is reached and the cycle ends altogether. What she decided to search for, then, were factors to help limit the amplitude. That is, factors to help contain both uncontrolled growth and uncontrolled decline in the sizes of the populations. In the course of further examining the model and its behavior, Talia devised a number of theories about the reason for the observed instability. She tested her theories through a process of successive revision, where she would repeatedly devise a corresponding variation to her rule set, instantiate it in NetLogo, and observe its effects.

Discussion: The danger of curve fitting. This activity of successive model revision is useful in that it allows students experience in developing original hypotheses, in formalizing them, and, to some degree, in testing them out. Modelers engaged in the process of model revision need to be aware of a potential danger: In attempting to alter a model to achieve a certain desired result, they run the risk of "curve fitting." That is, they may end up with a model that bears superficial similarity to the system that they are trying to model but achieves this using an unrelated mechanism. This danger occurs whenever there is a target behavior for a model, and once the target behavior is achieved, the model is not subjected to further testing in order to ensure a genuine correspondence. In general, embodied models are less prone to this danger than classical models, for they model systems at two levels (underlying mechanisms and global behavior) rather than just one (global behavior). This two-tier approach is safer, because there are more constraints that the modeler must satisfy (we elaborate on this in the "Discussion: Contrasting embodied versus classical assumptions" section). However, when modelers are not critical of the plausibility of their assumptions, the problem of superficial correspondence remains an acute danger. To avoid this hazard, students should always focus on the plausibility of the model as a whole rather than only on its behavior. For example, when the results of a model are not in line with expectations, a student should ask, what have I missed about the behavior of the components?—not, simply, how can I change my model to make it behave the way I want it to?

Researching the relevant biological literature. Research into scientific literature is often a part of the debugging process. This research can help amend any errors in a student's knowledge of the phenomenon or reveal any important facts that he or she might be overlooking. After experiencing difficulty devising a rule set that would lead to stable oscillations, Talia decided to do some research to determine the source of the problem. She discovered a substantial base of scientific literature addressing experimental evidence and theory of two-species predator–prey systems. Notably, she read that when such systems were first created in the laboratory by G. F. Gause (1934), the findings were just as with her NetLogo model: Either the predators ate all the prey and then starved, or, under certain conditions, the predators first died, and then the prey multiplied to the carrying capacity of the environment (Gause, 1934). Gause was surprised at this result. On the basis of the work of Lotka (1925) and Volterra (1926), he fully expected such two-species systems to be inherently stable. Talia thus learned, to her surprise, that her model was not necessarily wrong at all—it was the classical Lotka–Volterra model that was mistaken!

Talia's model failed to reproduce the dynamics of predator–prey systems found in nature, but it succeeded in predicting the dynamics that have been observed in the laboratory. Her research uncovered two important differences between the natural and the experimental settings to account for this discrepancy. The first is the lack of constraints on the growth of the prey population in the experimental settings. In nature, the size and rate of growth of the prey population are constrained by several factors, including limits on the food resources available to prey and limits on their maximum density. The laboratory experiments, however, included abundant food for the prey, and no other adversities in the system but the possibility of predation (Luckinbill, 1973). The second difference is the lack of environmental complexity—the models and the experiments leave no place for the prey to seek refuge and evade the predators, thus preempting the possibility of having some subpopulations of prey surviving in different regions (Huffaker, 1958).

Discussion: Contrasting embodied versus classical assumptions. Talia's model did not produce the expected results, but it turns out that this is only because her expectations were mistaken. The model omits any rules pertaining to environmental conditions or limits on food for prey, and thus it correctly predicts the outcome of the laboratory situation, which also omits these factors. The Lotka–Volterra model does not include these factors either, and given this, we would expect it to offer predictions for the experimental condition, not the natural condition. Indeed, Lotka (1925) and Volterra (1926) thought that their equations constituted a mathematical proof that such two-species predator–prey systems are inherently stable. Instead, this prediction has been shown false. Why might the two models differ in this way?

One might initially think that the different predictions offered by these two models can be attributed solely to skill (or luck) on Talia's part. In fact, neither skill nor luck can explain this alone—she tried very hard to achieve Lotka–Volterra-like behavior using NetLogo but was unable to do so without significant changes to her assumptions. These circumstances suggest that the factor that best accounts for her success versus Lotka's and Volterra's failure in this particular case was her use of embodied modeling tools.

Classical tools prevail in modern scientific practice because they provide, in many cases, an extremely concise and accurate representation of a system. Nevertheless, these tools must be applied with great care. Compared with embodied tools, classical tools make it much easier to model aggregate-level outcomes that are biologically implausible. Recall that classical and embodied tools each incorporate assumptions at different levels—the former at the aggregate level and the latter at the individual level. This is no small point. Whereas classical tools allow us to make any aggregate-level assumptions we want, embodied tools make it natural not to make any aggregate assumptions at all. Instead, we must code our assumptions at the individual level and wait to see what the aggregate-level consequences of these are. Depending on the outcome we have in mind, it may be that a reasonable individual-level rule set with this outcome simply does not exist.

It is still possible to make mistaken assumptions at the individual level, but there are two reasons why these may be easier to detect in embodied models than in classical ones. First, embodied models offer more feedback to the modeler; there are two levels at which to "debug" them, rather than only one level at which to debug classical models. We can scrutinize both the plausibility of the individual-level assumptions and the plausibility of the resulting aggregate-level outcomes. If either seems suspicious, then we have a hint that we may be on the wrong track. With classical tools, the only assumptions are, generally speaking, aggregate-level assumptions, so new information is not typically gained from observing a model's outcome.

The second reason why mistaken assumptions may be easier to detect in embodied models is that they take the form of rules for action. We have found that most students are already accustomed to thinking in terms of such rules, simply by analogy to their own experience. Hence, they come equipped with intuitive strategies for understanding and developing embodied models. For example, students will often try to make sense of a given rule set by assuming the perspective of the individuals within the model and using their imaginations (see Papert, 1980). Classical models, in contrast, require students to think in terms of more abstract quantities, such as rates and population densities. Although thinking in this mode may be comfortable for professional mathematicians, it is quite foreign to most students (see Stroup, 1996).

Discussion: Contrasting embodied versus classical explanations. Embodied models have another advantage over classical models that is particularly relevant in an educational setting. This advantage is that embodied models represent not only processes but also the mechanisms that underlie them. A classical model describes no more than a quantitative pattern: the Lotka–Volterra model describes a set of two curves. The explanation that it offers for these curves could be called a *shallow* one. It accounts for them by explaining that the birth rate of the predators is proportional to the number of prey and that the death rate of the prey is proportional to the number of predators. We use the term *shallow* because it is never actually specified how this explanation relates to actual organisms. In fact, this explanation can be induced just from looking at the population plots themselves—the very same plots we are trying to explain! Often when we ask for an explanation, though, we are looking for an underlying cause. That is, not an account of the pattern itself, but an account of the mechanism that gives rise to it. This deep kind of explanation, often more satisfying to students, is precisely what embodied models provide (Reisman, 1998; Wilensky, 1997b, 1999a). By bridging events with their underlying causes, deep explanations enable students to form powerful conceptual connections between their understanding of phenomena at different levels (see Wilensky & Resnick, 1999). Currently, most topics in biology (and in science, in general) are taught only at a single level. Be it the molecular, cellular, anatomic, organismic, or ecological level, these topics tend to be conveyed and understood in isolation from one another. It is unfortunate that the relations between these levels are not typically emphasized, given the possibility for topics at each level to provide deep, mechanistic explanations for topics at adjacent levels up. For example, students can apply their knowledge of molecules in order to make better sense of cellular processes and can apply their knowledge of organisms in order to make better sense of ecological processes. Not only do these applications provide a stronger intuitive basis for students to understand each topic, but they may also unify students' understanding of biology as a whole. Though computer tools are certainly not required in order to emphasize these conceptual connections between topics—many teachers already stress such connections to great effect in their lectures—our experience has shown NetLogo modeling to be a particularly effective means.

Adding grass—greater complexity can promote stability. Both the classical and embodied models presented earlier require emendation to account for the experimental findings. Many accurate classical models of predation have been developed since the work of Lotka and Volterra, but their mathematical complexity is beyond the scope of this article, and beyond the reach of most undergraduates (let alone high school students). We now turn to an alternative rule set that Talia devised to prevent her NetLogo ecosystem from destabilizing.

Talia learned that a major disparity between Gause's experimental setup and the natural case studies was the lack of constraints on the growth of the prey population. In natural systems, the prey population is generally constrained by the amount of resources available in the environment (e.g., food and living space) so that there is effectively a carrying capacity—a maximum number of organisms that can be supported—that limits the growth of the population. Gause's experimental setup and Talia's model both overlook this and instead include no such limits to growth. Prey within both systems have, at all times, ample food and ample space in which to live. As it turns out, surprisingly perhaps, this condition makes a significant difference to the stability of the system.

To impose a carrying capacity on the sheep population, Talia decided to modify her model so that sheep would now be required to consume some limited resource in order to survive. The new model would now include not only wolves and sheep but also grass, which would "grow" back once eaten. She represented the grass by means of patches that could either be green (i.e., grass is available for consumption) or brown (i.e., grass has already been consumed). Once a patch would turn brown, it would begin a countdown and only revert to green after some fixed interval of time. There were then two ways the prey could die—either by being eaten or by starving. These decisions resulted in an updated rule set for sheep and a new rule set for grass:

Rule Set S2: sheep

at each clock-tick:

1. move randomly to an adjacent patch and decrease energy by E_3
2. if on grassy patch, then eat "grass" and increase energy by E_4
3. if energy < 0 then die
4. with probability R_1 reproduce

Rule Set P1: patches

at each clock-tick:

1. If green, then do nothing
2. If brown, then wait X_1 clock-ticks and turn green

After selecting appropriate parameters and running her revised model, Talia found that her modifications had indeed brought about stable oscillations among the wolf and sheep populations. In addition, the level of grass in the model would oscillate as well. In examining plots of the population sizes over time, Talia noticed that changes in the sizes of the wolf population and in the level of grass would be roughly correlated, both varying as the approximate inverse of the number of sheep (see Figure 4).

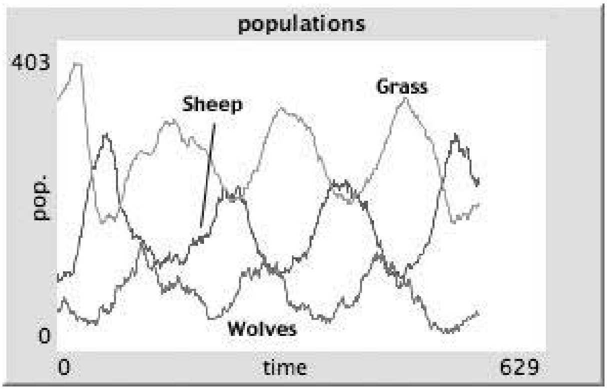

FIGURE 4 A typical outcome from Rule Sets W1 + S2 + P1. pop. = population.

In one sense Talia was surprised to find this model much more stable than the last, because she had given sheep more ways in which to die. Oddly enough, by limiting the resources of the sheep she had actually increased their chances of survival. On further reflection, Talia found this result—known in the literature as the "paradox of enrichment" (Rosenzweig, 1971)—entirely reasonable. By not controlling the amount of food available to the prey, the prey population can grow without limit. This increase in prey eventually causes the predator population to grow to unusually large levels, ultimately leading to a rapid and precipitous decimation of the prey. This effect is known to occur in the natural world just as it does in the NetLogo world. Accordingly, those involved in wildlife conservation efforts now know that providing endangered species with an excess of resources may have the counterintuitive effect of decreasing their numbers.

A further surprise for Talia was that the introduction of grass—an increase in the complexity of the model—actually contributes to stability. This model, with its multiple fluctuating and interdependent populations, is more reminiscent of an ecosystem than the previous version. Contrary to engineering logic, the logic of this model suggests that complexity and noise in a system can result in greater stability, not greater "chaos." Modelers in biology, both amateur and professional, are sometimes quick to abstract biological phenomena from the environments in which they occur. Talia's surprising result, however, urges us to remember that environmental and ecological context can play a significant role.

Spread Out the Sheep

Talia's revised model contains three species (wolves, sheep, and grass), although she originally set out to build a model of predation between only two (wolves and sheep). She wondered whether she could find other rule sets that would achieve the same effect as the grass, without including any additional species. She considered the role of grass in her revised model: It appears to ensure that only a finite number of sheep can inhabit a given area. If there are too many sheep, then the grass will run out and the sheep will starve, unless they move to another area that contains grass. Thus, she conjectured, the role of the grass is to limit the sheep population by placing a maximum density at which they can survive. Talia contemplated ways in which to impose such a density restriction on her model, without adding a third species. Another student, Benny, suggested that she could eliminate the grass and instead include a rule that explicitly restricts any patch from being occupied by more than one animal of a given species:

Rule Set S3: sheep
at each clock-tick:

1. move randomly to an adjacent patch which contains no sheep. If all adjacent patches contain sheep, remain in place.
2. if there is an unoccupied adjacent patch, then reproduce with probability R_1 and place the "offspring" into an unoccupied patch

Rule Set W2: wolves
at each clock-tick:

1. move randomly to an adjacent patch which contains no wolves. If all adjacent patches contain wolves, remain in place
2. decrease energy by E_1
3. if on the same patch as a sheep, then eat the sheep and increase energy by E_2
4. if energy < 0 then die
5. with probability R_1 reproduce

Benny and Talia hypothesized that the resulting dynamics of Rules W2 + S3 should look very similar to those of Talia's three-species model, only without the grass. In fact, when they first ran the model, they found its dynamics to be more similar to those of the first model Talia had built. The behavior was unstable, inevitably leading to the extinction of one or both of the two species, though it went through more oscillations before doing so. It was only after they spent considerable time varying the parameters of the model that they discovered an additional condition that promoted stability. They found that, by increasing the birth rates of the wolves and sheep to high levels, the sizes of the two populations would, indeed, continue to oscillate indefinitely.

Although W2 + S3 rule set results in stable behavior as the previous one did, a plot reveals that its dynamics are actually quite different (see Figure 5). Compared with the curves in the previous plot (see Figure 4), the two curves representing wolves and sheep in this plot are much more regular, are of far greater amplitude, and appear to have a different regular displacement from each other. It also shows,

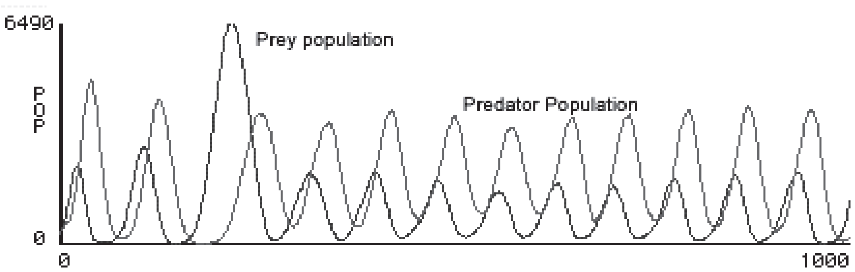

FIGURE 5 A typical outcome from Rule Set S3 + W2. POP = population.

unlike the previous plot, that the wolf and sheep populations are regenerated at each cycle from only a small number of individuals. Attention to the graphic window provides a dramatic manifestation of these differences; whereas the individuals in the previous model tend to be evenly dispersed and generally consistent in number, individuals in this model tend be more clustered in various regions of the window and subject to dramatic changes in number.

These two models are different, yet Talia could think of arguments in favor of each. The latter could be favored for its relative simplicity, as it contains only two different organisms (wolves and sheep) rather than three (wolves, sheep, and grass). Furthermore, its requirement that only one organism of a kind should be allowed per patch could be considered more realistic—why should an infinite number of organisms be allowed in a finite space? With the former model, Talia could respond that the patches do not represent small spaces but larger areas into which many organisms may fit. In addition, she could argue that this model's lack of dependence on a specific, and possibly unfeasible, range of birth rates makes it a more plausible alternative.

How should Talia have chosen between these competing rule sets? Which one should she have deemed correct? Her answer was that one cannot choose between equally plausible rule sets so long as they both yield equally plausible results. She came to see that her difficulty in choosing was not specific to this example or to NetLogo modeling—indeed, it is inherent in the process of scientific modeling. When multiple theories are equally compatible with existing knowledge, and neither theory is more predictive than the other (as in this case), then there will be no direct way to arbitrate between them (Quine, 1960).

Discussion: Answers versus theories. Some teachers within the MSCP project were initially uncomfortable with this indeterminacy and sought to hide it from students. In our discussions with these teachers, we encouraged them to "dive into" the indeterminacy. Not only is such indeterminacy fundamental to scientific inquiry, but it may be valuable to students as part of their own thought processes.

The modeling-oriented approach to learning biology shifts their goal from finding the correct theory to finding a theory that is compatible with all the available evidence. The significance of this shift is that students no longer need to search only for unique answers, which may be true or false in themselves. They can spend their time trying to compare theories against other theories. This shift of focus for the student—from learning answers to assessing theories for themselves—is just the kind of high-level skill called for by educational policymakers and industry leaders at a time in which the turnover of scientific knowledge is so rapid (Chen, 1999; Murname & Levy, 1996). Although content knowledge, or many of the "answers" in today's textbooks, is already out of date, the skill of assessing the validity and plausibility of answers is not so easily made obsolete.

Before we move on to the next example, it is worth noting that the methods and tools we have presented here are useful not only to students but to professional scientists as well. The embodied approach, known to population biologists as *individual-based modeling*, has become an accepted methodology within the field. Current individual-based models of predation are actually rather similar to the one we have developed in this section. In some cases, these models offer greater predictive accuracy than their classical counterparts (Huston et al., 1988; Judson 1994).

MODELING SYNCHRONIZED FIREFLIES

In the previous section, we showed how students can use NetLogo as a tool to model and explore biological systems. In this section, we elaborate on the ways in which students can, through the process of modeling, both learn about specific topics within biology and use the NetLogo modeling language as a laboratory for exploring biological mechanisms. Our example follows the inquiry of an undergraduate student, Paul, whose formal biology instruction consisted solely of high school biology courses. Through his involvement with the MSCP project, Paul learned of the phenomena of synchronously flashing fireflies and was intrigued. The following paragraph provides some background.

For centuries, travelers along Thailand's Chao Phraya River have returned with stories of the astonishing mangrove trees that line its banks. Come nightfall, these trees have been seen to flash brilliantly, on and off, intermittently illuminating the surrounding woods and the water below. A closer look at this display, though, reveals that the sources of these rhythmic flashes are not the trees at all. Rather, it the effect of thousands of individual fireflies inhabiting the trees, all pulsing their lights in unison. Several species of firefly are known to do this, such as the Southeast Asian *Pteroptyx malacae* and *Pteroptyx cribellata*. When one such firefly is isolated, it will typically emit flashes at regular intervals. When two or more such fireflies are placed together, they entrain to each other—that is, they gradually converge upon the same rhythm, until the group is flashing in synchrony (Buck, 1988).

How do the fireflies achieve this coordinated behavior? When we think about how behavior is coordinated in our daily lives, we tend to think of schedules and elaborate plans. Paul was perplexed at how creatures that seem to have little capacity for such intelligent planning are nonetheless capable of such coordination. It was Paul's suspicion that there must be a simple mechanism behind the feat of the synchronizing fireflies. His goal was to try to understand this mechanism by building a model of it in NetLogo. Paul decided not to begin his inquiry by doing an extensive literature search. Instead, he was determined to see if, perhaps, he could find a solution on his own. He began his task with no more than the description just given.

Approaching the Problem—Initial Assumptions

To begin, Paul made several working assumptions about these fireflies; he was prepared to revise them later if necessary. First, he decided that the mechanism of coordination was almost certainly a distributed mechanism. That is, the fireflies were not all looking to a leader firefly for "flashing orders," but rather were achieving their coordination through passing and receiving messages from other fireflies. From his previous experience with NetLogo, he had learned that not all coordinated group behavior requires a purposeful leader to direct the group (see Resnick, 1996; Wilensky & Resnick, 1999). Examples such as the food-seeking behavior of ants and the V-flocking of birds implied that some forms of group organization could arise on their own. That is, as long as each organism follows a certain set of rules, then the whole group would be likely to organize itself. Paul's seeking out of a distributed mechanism to explain firefly synchronization represents no small achievement.[13] Elsewhere (Resnick, 1996; Resnick & Wilensky, 1993; Wilensky & Resnick, 1995, 1999), we have described a "deterministic–centralized mindset"—a tendency of most people to describe global patterns as being orchestrated by a leader giving deterministic orders to his or her followers. Paul's experience in the MSCP project allowed him to overcome this tendency and consider leaderless nondeterministic mechanisms for firefly synchronization. Given the limited intelligence of individual fireflies, Paul surmised that just such a mechanism probably underlies firefly synchronization behavior. A second assumption, following the first, was that the system could be modeled with only one set of firefly rules—that is, with every firefly in the system following the same set of rules. Although he recognized that this assumption might have been too strong, just as ant and bee populations do divide roles among their groups, he decided to first try out the simpler hypothesis of undifferentiated fireflies. Yet a third assumption Paul made concerned the movement of the fireflies—that it was not necessary to model this movement as coordinated or governed by deterministic rules but rather it could

[13] It is interesting to note that some of the first theories proposed to explain synchronously flashing fireflies were, in fact, "leader" theories (Hudson, 1918; Morse, 1916).

be modeled as random flights and turns. From experience with other NetLogo models, he had come to appreciate the role of randomness in enabling coordination (Wilensky, 1997b, 1999a). In a wide variety of domains, ranging from the movements of particles in a gas to the schooling of fish and the growth of plant roots, Paul had seen how stable organization could emerge from nondeterministic underlying rules. A final assumption was that the behavior of the fireflies could be modeled in two dimensions.[14]

Beginning With a Simple Model

These assumptions left Paul with the task of finding a plausible set of rules for a typical firefly. Rather than tackle this problem all at once, he decided it would be easier to begin with a simpler version. He started by modeling a flashing firefly that does not synchronize. Paul contemplated how to represent a flash using a NetLogo turtle. The solution came naturally: The turtle would change its color (say, to yellow) and then change it back. Now he needed a mechanism to regulate when the flash would occur. He knew that if left alone, a firefly would continue to emit flashes at a constant rate. Paul considered how to represent this simple behavior within his model firefly; the behavior meant that every several clock-ticks, the model should flash (change its color). For the flash to be seen, it would have to last at least one clock-tick. To accomplish this goal, Paul decided to give the model firefly a timer that would count down from a predefined reset-value (R)—once the timer reached zero, the firefly would flash and reset the timer. In addition to the flash-timer, Paul also included a rule to cause the model firefly to "fly" around the screen. He assumed provisionally that a randomly generated flight path would be sufficient. He wrote the following rule set:

```
Rule Set F1: firefly
to initialize:
  0. set timer with random value between 0 and R

at each clock-tick:
  1. if color is yellow (flash is on), then change color to black (flash is off)
  2. if timer is zero, then change color to yellow and reset timer to R
  3. decrement countdown timer by one
  4. move randomly to an adjacent patch
```

[14] While the first three assumptions were derived, to a great extent, from Paul's understanding of plausible biological mechanisms, this last assumption was primarily driven by the limitations of computer displays and of the NetLogo language itself. Because building a three-dimensional model was more difficult in that version of NetLogo, Paul was, essentially, hoping that the three dimensionality of the firefly world was not the key factor in enabling their coordination.

After Paul debugged his NetLogo code, his simple model worked. The fireflies would move around the screen and flash regularly, though of course they did not yet synchronize their patterns.

Thinking Like a Firefly

Paul was left to ponder what sort of additional rules might cause the fireflies to synchronize with each other. He considered the nature of coordination in general: Could it ever be possible for distinct entities to coordinate their behaviors if they were unable to communicate with each other? No—it seemed communication at some level would always be necessary for any coordinated behavior to occur consistently. He wondered what kind of communication mechanism might be used.

Often when building a model, students find it helpful to identify with the individuals within the model and to view phenomenon from their perspective. At this point, Paul began to "think like a firefly." He reasoned along the following lines: If I were a firefly in that situation, what information would *I* have to go on? It would be dark, and I probably would not be able to see the other fireflies. I probably would not have much capacity for hearing or sensing the other fireflies either. I would, however, be able to see their flashes. Perhaps, then, I could look to see who else is flashing and then use this information to adjust my own flashing pattern.

Sorting Through Design Options

Paul concluded that the flashes themselves could serve to communicate the necessary information, and he wanted to make this possibility more concrete within the context of the NetLogo environment. He had already decided that in order to flash, a firefly changes its color from black to yellow and back to black again. A firefly must, then, be searching for other yellow fireflies. There are many ways that such a search might be carried out in NetLogo, and Paul found that he had some choices to make. The process of formalizing his model forced Paul to confront questions that he had not already considered: How many other fireflies should a firefly look at? At what distance could it detect a flash? How many flashes should it be allowed to take into account?

Paul saw that there did not have to be any strictly correct answers to these questions, because they were questions about simulated fireflies, not actual fireflies. For example, it would make little sense to ask how many patches away actual fireflies can see! Still, Paul thought that at least the issue of whether a model firefly should survey the flashes of all or only a part of the population should have a clear answer. For it would be possible to allow a model firefly to detect all flashes in the population at a given clock-tick, but this would surely be granting the firefly too much information; model fireflies should not be much more intelligent or perceptive than real ones. Because a real firefly would only be able to perceive a subset of the flashes in the population, Paul decided that the model firefly should scan only adjacent patches to look for yellow fireflies. He began with a rule that allows a firefly to sense other flashes within a radius of one patch. This decision only partly simplified the question of what a firefly senses. Consider some statistics that a firefly might collect about observed flashes: overall brightness (i.e., the combined light of all observed flashes) during a given clock-tick, the number of distinct flashes observed during a clock-tick, the number of clock-ticks between observed flashes, increases in relative brightness from clock-tick to clock-tick, simply whether or not any flash had been observed at all at a given clock-tick, and so on. At this time, Paul did not have any principled way of choosing among these data-collecting options, so he decided to proceed without committing to any of them.

Given that a firefly has some mechanism for perceiving flashes, and perhaps for analyzing this information in some way, the next question that Paul faced was what to do with this information. In what way would a firefly alter its flashing behavior in response to whatever had been observed? Paul tried to think of a simple situation to make sense of the problem. Once again, he took the perspective of a firefly: Suppose I perceive a clear pattern among the other fireflies—for example, everyone else is already synchronized. Then, as long as we all have timers of the same duration, it would be simple to match this pattern. On seeing everyone flash, I would reset my timer as if I had flashed as well. Then my next flash would coincide with everyone else's.

Having understood what to do at one extreme, Paul tried to work backwards: At some point, before everyone else is synchronized, I must be confronted by a multitude of unsynchronized flashes. Then what would I look at? To what would I reset my timer? Paul thought again of all the different ways that a firefly might analyze observed flashes and all the different timer-reset rules that must be possible. Paul had many ideas, but he felt that he needed more information to continue.

Researching the Relevant Biological Literature

Notice how far Paul was able to get without reference to detailed information about the real-world phenomenon. From his initial goal to model "whatever" was going on, by "thinking like a firefly," he was able to reason to this point where he was seeking a very particular sort of algorithm. It might have been possible for someone to take this line further, but to decide among some of the options he had left open, Paul felt a need to gather information about the behavior of real fireflies (e.g., Does the interval between successive flashes vary from firefly to firefly? How far can an actual firefly see? How many flashes can it take into account? What is the timer-reset rule?).

At this point, Paul did some research into the scientific literature. His own investigation had not answered all his questions, but it had given him a sound context from which to understand and interpret the existing research. In looking through the literature, he was not reading a teacher's assigned material but rather engaging in his own research to answer questions of his own devising. Paul located several journal articles to help answer his questions. He found out the following (Buck, 1988; Buck & Buck, 1968, 1976; Carlson & Copeland, 1985):

1. Fireflies do indeed have internal "timers" to regulate the period of flashing and are known to entrain their timers to observed rhythms.
2. The only information needed for entrainment is that of other flashes.
3. There are many species of synchronizing fireflies, all with different entrainment and oscillation characteristics.
4. Some species are able to adjust the period of oscillation; others are not. For the latter, the period of oscillation tends to remain highly constant across a population.
5. Different methods of synchronization are seen across species. Two main mechanisms are phase delay and phase advance synchronization:
   - (a) *Phase-delay*: When a firefly perceives *any* flash, it delays its next flash so that it will occur one period after the perceived flash. This strategy is known to be used by the Southeast Asian fireflies of the *Pteroptyx* genus.
   - (b) *Phase-advance*: When a firefly perceives any flash during a short window of time before flashing, it flashes immediately and starts a new period immediately thereafter. This strategy is known to be used among species that tend to synchronize only rarely and transiently, such as the American *Photinus pyralis*.

Paul was pleased to discover that many of his design decisions were biologically plausible, such as his focus on a distributed synchronization mechanism, his use of timers to control flashing, and his decision to allow timers of the same duration across the population. The synchronization mechanism he had thought of earlier appeared to correspond to the phase-delay mechanism from the text, although he was surprised to learn that a firefly needs to see only one flash, any flash, to react. His next step was to extend his existing model in order to determine whether this would really work.

Modeling Phase-Delay Synchronization

Paul decided to model the phase-delay mechanism first. The research did not turn up any information on the maximum distance within which a firefly can perceive other flashes, and therefore, Paul had to decide on this matter on his own. For representational simplicity, he chose to allow fireflies to sense other fireflies within a radius of one patch. Incorporating this information in his earlier model resulted in the following rule set:

Rule Set F2: phase-delay firefly
0–4. Identical to rule set F1
5. if there is a yellow firefly within one patch, then reset the timer to R

Paul ran Rule Set F2 using 1,000 fireflies and was amazed to see the model fireflies converge on a single rhythm before his eyes. He also set up a plot to display the number of fireflies flashing at a given time (see Figure 6).

Modeling Phase-Advance Synchronization

Next, Paul wanted to try out the phase-advance strategy. This attempt required more sophistication, because a phase-advance firefly will only adjust its timer during a short window before its flash. Paul amended F2 to account for this time window:

Rule Set F3: phase-advance firefly
0–4. Identical to rule set F1
5. if there is a yellow firefly in a neighboring patch, and I am within W clock-ticks of my next flash, then reset the timer to R

Indeed, Paul found that this strategy was not as effective as phase-delay synchronization—when he ran this rule set (see Figure 7), he did not observe synchronization at all! It was only after much experimenting that he discovered a variant of this rule set (see F4) that did produce synchrony, although the synchrony took much longer to develop and was not as precise as with Rule Set F2 (see Figure 8). Rather than flash in perfect unison, the fireflies would all flash within an interval of two or three clock-ticks.

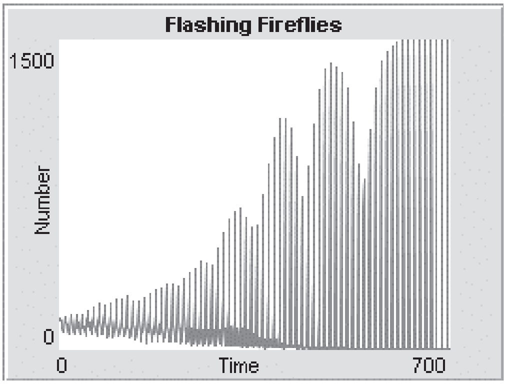

FIGURE 6 Typical plot of the number of flashes in a firefly population at a given time under Rule Set F2.

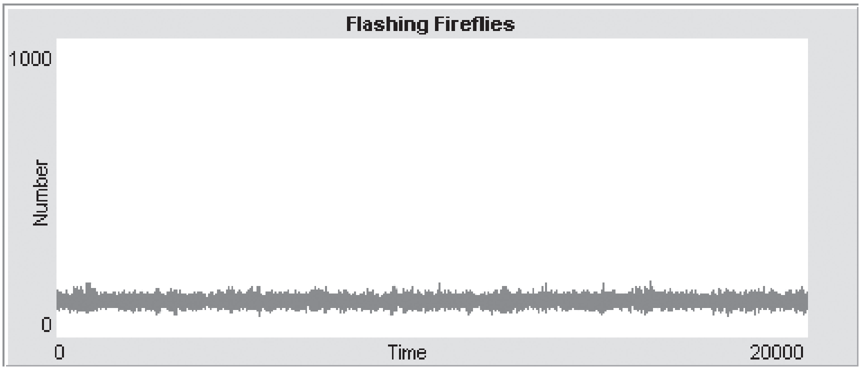

FIGURE 7 A typical result of running Rule Set F3. Even after 20,000 clock-ticks, no synchrony emerges.

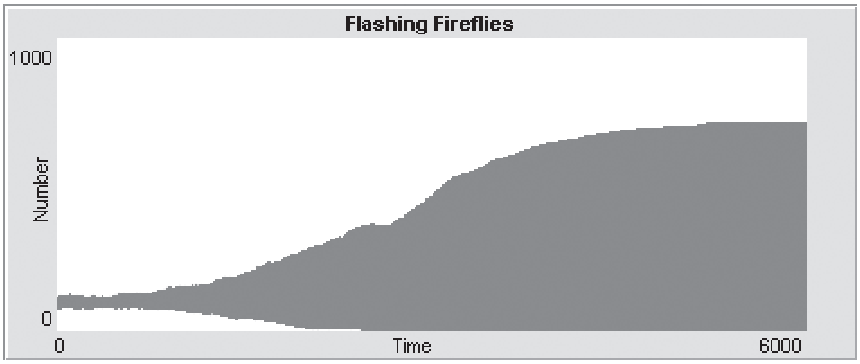

FIGURE 8 A typical result of running Rule Set F4. Fireflies eventually synchronize, but not with the same speed or precision as in Rule Set F3.

Rule Set F4: phase-advance firefly

0–4. Identical to rule set F1

5. if there are *at least two* yellow fireflies within one patch, and I am within W clock-ticks of my next flash, then reset the timer to R

Although he had managed to achieve synchronization using a phase-advance mechanism, Paul was uncomfortable with this result as he had done so by means of an ad hoc change to his model. He wondered whether, perhaps, this result predicted the behavior of actual fireflies, or whether it was a just an artifact of the representational decisions he had made.

Paul tried a number of further variants to Rule Set F4 to investigate other possible flash-reset mechanisms. Among them was one where he omitted the requirement for a flash-window (W). He quickly discovered why this window was necessary: Without it, the fireflies would persistently reset each other's timers, and there would be no interval between flashes.

Further Questions for Research

Paul was encouraged by the initial results of his research and was left with new questions to investigate. For example, he was intrigued by the ability of some fireflies to adapt not only the timing of their flash, but also the duration between flashes. The publications he had looked at gave no complete theory of how this could be done. He was also interested in customizing his model to reflect the idiosyncrasies (e.g., multiple consecutive flashes, responses to irregular stimuli) of particular species, such as *Pteropox malacae* and *Photinus pyralis*. Though he began his inquiry with only a single question in mind, he found that his questions multiplied as his research continued.

Discussion: The Model Testing Process

What makes a model a scientific model is that it has been tested against whatever system it was designed to represent. At this point, Paul's phase-delay model was successful in having the model fireflies collectively synchronize their flashing patterns, but the correspondence between this model and reality still had to be tested. Indeed, Paul went through such a process of testing and eventually was convinced of the soundness of his modeling decisions. In this section, we remark on his experience frame a discussion about the process of testing and evaluating NetLogo models in general.

Why bother testing at all? In one sense we know that Paul's model works, for the model fireflies do indeed synchronize. However, we have already discussed the dangers of models that bear only a superficial correspondence to the phenomena being modeled, and in Paul's case there is certainly such a danger. Perhaps there are many different algorithms that will lead to synchronization. How do we know that Paul's model corresponds in a meaningful way to the behavior of real fireflies? We have some indication, because journal articles have confirmed many of Paul's assumptions. Still, Paul may have made errors in coding his model, he may have misinterpreted the literature, and, of course, the literature itself may have been incorrect. Ideally then, we would like more evidence that the model is sound. To evaluate a model, a modeler must critically analyze the content and output of the model along several dimensions. These include the soundness of the model's underlying assumptions, the robustness of its output, and its predictive capacity. Let us consider each of these in turn with respect to Paul's model.

Continually evaluating the plausibility of the background assumptions is a modeler's first line of defense against specious models. Whenever design decisions are made, the modeler should be aware of the ways in which these decisions may detract from the realism of the model. He or she may then make a deliberate choice to stay with these decisions, reject them, or, perhaps, to wait until later to choose. For example, when Paul decided that all fireflies would follow the same rules or that the relevant behavior could be adequately modeled in two dimensions, he made these decisions provisionally. He was aware of the conceptual jump he had made and was prepared to retract these assumptions if necessary. Any model will take representational liberties. The important thing is to be aware of these, and to try and discern whether and how they affect the plausibility of the model.

Another way to evaluate a model is to consider its "robustness." A robust model will yield consistent results, even when we introduce noise, adjust the parameters, or even effect small changes to the background assumptions. If we do obtain consistent results under these conditions, we have evidence that our model is not overly sensitive to our assumptions or chosen parameters—some of which may be arbitrary or mistaken. We may reasonably suspect a nonrobust model of being implausibly contrived, or curve fitted (see the earlier "Discussion: The danger of curve fitting" section). It was on these grounds that Paul had been suspicious of his phase-advance model. He wanted to further test his phase-delay model along the same line. Paul figured that a robust solution to the synchronization problem should be able to hold up under nonideal, or "noisy," conditions, where other factors might interfere with the phase-delay algorithm. One way that he tested his solution was to introduce several "blind" fireflies that would not synchronize with the rest but would still flash. When he tried this, he found that the population took much longer to settle into a stable pattern of synchronization. Initially, only small clusters of synchronization would transiently form and then break up again within the population (see Figure 9). Paul was pleased to find that the algorithm did hold

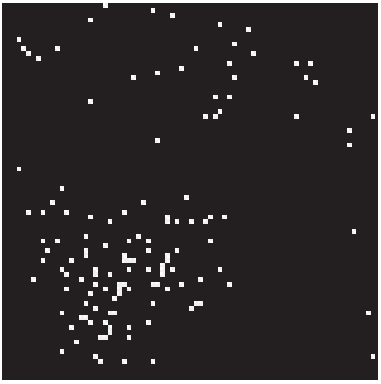

FIGURE 9 Clusters of synchronization within a population of 1,000 fireflies.

up under this test condition. He then found in the literature that this local clustering had even been observed in natural *Pteroptyx* populations (Buck, 1988).

Of course, the principal way that scientific models are evaluated is by determining how well they can be used for prediction. By this, we mean that the model anticipates some result that is approximately true of the system being modeled, and that did not itself factor into the development of the model. New data against which to compare a model might be collected in the laboratory or from nature, though research in journals and other texts will often provide enough data for the purposes of students. The predictions that student models may offer for this data are typically of a qualitative rather than quantitative nature. In Paul's case, for example, there was the unexpected result from the "noisy" fireflies. His later discovery that this is actually in accord with real firefly behavior constitutes an item of predictive evidence in favor of his model. Ideally, a student would attempt to find and amass as much such evidence as possible.

Testing can take place after students have developed their initial models, or, often, it will be concurrent with the process of development. All along, students always have the options of either concluding the modeling process or going back and revising their models in light of what they have discovered. In the end, even after students have critically evaluated their models, they must (once again) confront the inevitable indeterminacy that surrounds the testing of scientific theories. Theories are never conclusively proved (Popper, 1959). Accordingly, students should not walk away from the modeling process believing themselves to have found the correct solution. Rather, they should leave with an awareness of the ways that their model both does and does not reflect the system they set out to capture.

Critical thinking about modeling does not come easily to many students. In the MSCP project, we have observed that for many students NetLogo modeling is their first experience where such thinking—the sort that underlies experimental science—is demanded. We have found that students engaged in NetLogo modeling, through revising, assessing, and successively refining their models do, indeed, develop a propensity for critically evaluating their models. This propensity, however, is hard won. Typically, it is only after a good deal of guidance that students will become critical of the representational decisions they have made. Further research is needed on how to help students to move beyond good model building to good model critique. In our concluding remarks, we argue that significant learning occurs even when students do relax the requirement of criticality.

Further Reflections: Learning Through Building

Let us call this the engineer's dictum: If you can't build it, then you don't understand it. Our approach of modeling underlying mechanisms takes the engineer's dictum seriously. To model a system, it is not sufficient to understand only a handful of isolated facts about it. Rather, one must understand many facts and concepts about the system and, most important, how these relate to each other. The process of modeling is inherently about developing such conceptual relations and seeking out new facts and concepts when a gap in one's knowledge is discovered.

We have seen how Paul came, through building, to understand the concept of a simple circuit capable of entrainment. In science and mathematics, such circuits are known as oscillators, and networks of such circuits are known as coupled oscillators. As it turns out, such oscillators underlie not only firefly synchronization but also a wide-range of phenomena throughout biology that exhibit synchronization behavior without any centralized control. Among other phenomena, oscillators are involved in acoustic synchronization between crickets, the pacemaker cells of the heart, neural networks responsible for circadian rhythms, insulin-secreting cells of the pancreas, and groups of women whose menstrual periods become mutually synchronized (Mirollo & Strogatz, 1990). Though Paul's goal was to learn about fireflies, he had come to understand a concept that has applications far beyond.

CONCLUDING REMARKS

The embodied modeling approach we have presented and illustrated herein makes practical a modeling-centered biology curriculum in secondary and postsecondary contexts. By removing the barriers of formal mathematical requirements, it enables students to meaningfully engage the dynamics of complex biological systems.[15] They are able to construct models of such systems, reason about the mechanisms that underlie them, and predict their future behavior. Because they are able to use their knowledge of the individual elements in the system to construct their model, they are provided with an incremental path to constructing robust models. When their knowledge of the individual biological elements is combined with their knowledge of their own embodiment, their own point of view, they are enabled to think like a wolf, a sheep, or a firefly.

Thinking Like a Scientist

The examples described here have, we hope, demonstrated the power of the embodied modeling approach to enable students to construct robust models and engage in exciting scientific inquiry. For some readers, there may still remain the question of why any kind of modeling approach should be given a significant share

[15] Elsewhere (Wilensky, 1995, 2001) we have argued that computational modeling approaches are a new alternative form of mathematics—a new mathematics for a new way of describing, representing, and investigating scientific theory (see also Abelson & diSessa, 1981; Noss, 1988; Noss & Hoyles, 1996; Papert, 1972, 1980).

of classroom time. We conclude by mounting a defense of a general modeling approach in the science and mathematics classroom.

The modeling-based classroom is dramatically different from most venues of classroom practice. Rather than passively receiving an authority's explanation of science and mathematics concepts, students seek out and consider these concepts on their own. Rather than carry out the directions for predetermined lab studies, students engage in new investigations. What underlies this approach is our deep conviction of the value of reasoning about scientific order. In both the predation and firefly examples presented in this article, students were encouraged to reason through a problem, creating and testing their own theories and hypotheses, before reaching for the established literature.

A critic of our approach might argue that students may be prompted to develop and teach themselves false models. We have already emphasized the importance of encouraging a critical analysis of all models to avoid such false solutions. However, we acknowledge that given the theoretical level at which we encourage students to consider problems, it is not unlikely that students will indeed develop models that are at variance with natural systems. It is important to note that we do not believe that this is a problem. Let us explain.

Methodology aside, educators differ about the goals of secondary and undergraduate science education. Some common views are (a) to convey knowledge of specific scientific facts and techniques, (b) to foster in students a general understanding of and appreciation for the world around them, (c) to train students in tools and approaches that will prepare them to learn about and assess scientific theories they have not previously encountered, and (d) to prepare students to develop their own theories and conduct their own scientific research. No doubt, educators may value several or all of these objectives; indeed, we believe they are all important. The distinctive form of our approach, which emphasizes independent consideration of scientific topics, responds to our belief that none of the aforementioned objectives are adequately met by the standard science curriculum alone.

Very often, science classes effectively amount to tests of students' abilities to memorize large numbers of facts. Sometimes, the classes manage to emphasize an intuitive understanding of these facts within a larger context. However, rarely are the other two objectives even attempted, let alone emphasized; general scientific methods and processes of thinking are generally overlooked. This omission is due to many different factors, including the difficulty in constructing tests that assess such processes, the pressure to achieve broad coverage of the curricular topics, and the discomfiture caused to teachers and school administrators by the change in teacher and student roles in a modeling-centered curriculum. In our view, teaching scientific facts without placing these within a larger context—which includes conveying how this knowledge was established and how new scientific information comes to light—misses the point. This miss is why, above all, the modeling approach we have presented here emphasizes a process rather than a result. Regardless of one's educational priorities, it is a mistake to assume that one can achieve the first objective listed while dropping the last three of these objectives. Particular facts and theories must have a context of processes and beliefs to be integrated with existing knowledge and retained. This sense-making context is all the more important for those students who will not continue in the study of science and for whom the isolated facts remain "one-night stands."[16]

We can now return to our assertion that we do not take the possibility of students teaching themselves false models to be a major problem. We have argued here that it is more important to convey to students general methods, notions, and processes of thinking than it is to emphasize specific theories—at least at the secondary and early undergraduate levels. A consequence of this decision is that we have to relax (not drop) our insistence on correct answers. Students will not learn to be rigorous scientists overnight. They will generally need to go through a process of exploring and experimenting with the techniques and ideas we have discussed before these become natural to them. Yet, if we penalize them each time they express ideas that are strictly incorrect, we are sure to stifle their motivation for such creative exploration.

Our approach promotes several processes of reasoning that are central to science: developing original hypotheses, formalizing ideas, researching existing solutions, and critical analysis of results. We believe that experience with these processes will be of significant advantage to all students as they seek to understand science and, more generally, the world around them. Few students will go on to become scientists. To the ones that do not, we, as educators, owe more than just an introductory glimpse of current theories—we owe them the tools with which to appreciate scientific evidence and to engage in scientific inquiry for themselves. To the ones that do go on to become scientists, we owe a framework within which they will be better prepared to absorb and appreciate the myriad facts they will encounter for years to come. Thus, it is our hope that the approach we are developing will serve as a framework for all students. We believe it is critically vital for both future scientist and future nonscientists–citizens to be able to work and think like scientists.

[16] Deanna Kuhn (1993) made this point eloquently: "Scientific thinking tends to be compartmentalized, viewed as relevant and accessible only to the narrow segment of the population who pursue scientific careers. If science education is to be successful, it is essential to counter this view and establish the place that scientific thinking has in the lives of all students. A typical approach to this objective has been to try to connect the content of science to phenomena familiar in students' everyday lives. An ultimately more powerful approach may be to connect the process of science to thinking processes that figure in ordinary people's lives" (p. 333).

ACKNOWLEDGMENTS

The preparation of this article was supported by Grants RED–9552950 and REC–9632612 from the National Science Foundation. The ideas expressed here do not necessarily reflect the positions of the supporting agency.

A condensed version of this article appeared in the proceedings of the Second International Conference on Complex Systems, Nashua, NH, October 1998. As a considerable period elapsed between the time this article was accepted (1999) and its publication, some recent relevant work has not been referenced that would have been included if the article was written at this time.

We thank Seymour Papert for his overall inspiration and support and for his constructive criticism of this research in its early stages. Mitchel Resnick has been an invaluable collaborator in thinking about issues of complexity and education in general and in exploring the predator prey models in particular. Steve Longenecker and Josh Mitteldorf contributed greatly to our thinking about the predator–prey models. Ed Hazzard, Walter Stroup, and Rob Froemke made valuable clarifying suggestions. Daniel Dennett provided valuable criticism of the firefly model and inspired many of our observations about embodied modeling. Many people contributed significantly to the development of the NetLogo and StarLogoT environments. Rob Froemke played a significant role in the development of the StarLogoT modeling environment with the assistance of Eamon Mckenzie, Paul Deeds, and Dan Cozza. Mitchel Resnick, Brian Silverman, and Andy Begel were largely responsible for a previous StarLogo version, MacStarLogo, on which StarLogoT was based. Seth Tisue played a significant role in the development of the NetLogo modeling environment with the assistance of Matt Goto and many others. Eric Betzold and Josh Unterman provided needed assistance with the figures. We wish to thank Rich Lehrer, Dor Abrahamson, Michael Jacobson, Bill Rand, and the anonymous reviewers who made many valuable suggestions to improve the final document.

REFERENCES

Abelson, H., & diSessa, A. (1981). *Turtle geometry: The computer as a medium for exploring mathematics.* Cambridge, MA: MIT Press.

Allen, G. (1975). *Life science in the twentieth century.* New York: Wiley.

Beer, R. D. (1990). *Intelligence as adaptive behavior: An experiment in computational neuroethology.* Cambridge, MA: Academic.

Buck, J. (1988). Synchronous rhythmic flashing of fireflies II. *Quarterly Review of Biology, 63,* 265–287.

Buck, J., & Buck, E. (1968, March 22). Mechanism of rhythmic synchronous flashing of fireflies. *Science, 159,* 1319–1327.

Buck, J., & Buck, E. (1976, May). Synchronous fireflies. *Scientific American, 234,* 74–85.

Buneman, O., Barnes, C., Green, J., & Nielson, D. (1980). Principles and capabilities of 3-E-M particle simulations. *Journal of Computational Physics, 39,* 1–44.

Carey, S. (1986). Cognitive science and science education. *American Psychologist, 41,* 1123–1130.

Carlson, A. D., & Copeland, J. (1985). Flash communication in fireflies. *Quarterly Review of Biology, 60,* 415–436.

Chen, D. (1999). Confronting complexity: A new meaning to world literacy. In E. Chaisson & T. C. Kim (Eds.), *The thirteenth labor: Improving science education* (pp. 87–100). Amsterdam: Gordon & Breach.

Dennett, D. (1995). *Darwin's dangerous idea: Evolution and the meanings of life*. New York: Simon & Schuster.

Elton, C. (1966). *Animal ecology*. New York: October House.

Epstein, J., & Axtell, R. (1996). *Growing artificial societies: Social science from the bottom up*. Washington, DC: Brookings Institute.

Feurzeig, W. (1984). The Logo lineage. In S. Ditlea (Ed.), *Digital deli* (pp. 158–161). New York: Workman.

Feurzeig, W., & Roberts, N. (1999). *Modeling and simulations in science and mathematics education*. New York: Springer-Verlag.

Forrest, S. (1989). *Emergent computation: Self-organizing, collective, and cooperative phenomena in natural and competing networks*. Amsterdam: North-Holland.

Gause, G. F. (1934). *The struggle for existence*. Baltimore: Williams & Wilkins.

Gobert, J., Horwitz, P., Tinker, B., Buckley, B., Wilensky, U., Levy, S., et al. (2004). Modeling across the curriculum: Scaling up modeling using technology. In R. Alterman & D. Kirsh (Eds.), *Proceedings of the Twenty-fifth Annual Meeting of the Cognitive Science Society* (p. 1349). Boston: Cognitive Science Society.

Hammer, D. (1994). Epistemological beliefs in introductory physics. *Cognition and Instruction, 12,* 151–183.

Hatano, G., & Inagaki, K. (1987). Everyday biology and school biology: How do they interact? *The Quarterly Newsletter of Comparative Human Cognition, 9,* 120–128.

Hudson, G. (1918, December 6). Concerted flashing of fireflies. *Science, 48,* 573–575.

Huffaker, C. (1958). Experimental studies on predation: Dispersion factors and predator–prey oscillations. *Hilgardia, 27,* 343–383.

Huston, M., DeAngelis, D., & Post, W. (1988). New computer models unify ecological theory. *BioScience, 38,* 682–691.

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

Jacobson, M., Brecher, K., Clemens, M., Farrell, W., Kaput, J., Reisman, K., et al. (1998). Education in complex systems. In Y. Bar-Yam (Ed.), *Proceedings of the Second International Conference on Complex Systems* (pp. 549–556). Nashua, NH: New England Complex Systems Institute.

Jacobson, M., & Kozma, R. (2000). *Innovations in science and mathematics education: Advanced designs for technologies of learning*. Mahwah, NJ: Lawrence Erlbaum Associates, Inc.

Judson, O. (1994). The rise of the individual-based model in ecology. *TREE*, *9*(1), 9–14.

Jungck, J. R., & Calley, J. (1985). Strategic simulations and post-Socratic pedagogy: Constructing computer software to develop long-term inference through experimental inquiry. *American Biology Teacher, 47,* 11–15.

Kauffman, S. (1995). *At home in the universe: The search for the laws of self-organization and complexity*. Oxford, England: Oxford University Press.

Keeling, M., & Gilligan, C. (2000, October 19). Metapopulation dynamics of bubonic plague. *Nature, 407,* 903–906

Keen, R., & Spain, J. (1992). *Computer simulation in biology*. New York: Wiley-Liss.

Kuhn, D. (1993). Science as argument: Implications for teaching and learning scientific thinking. *Science Education, 77,* 319–337.

Langton, C. (1994). *Artificial life III*. Reading, MA: Addison-Wesley.

Lehrer, R., & Schauble, L. (2000). Modeling in mathematics and science. In R. Glaser (Ed.), *Advances in instructional psychology: Vol. 5. Educational design and cognitive science* (pp. 101–159). Mahwah, NJ: Lawrence Erlbaum Associates, Inc.

Lotka, A. J. (1925). *Elements of physical biology.* New York: Dover.

Luckinbill, L. (1973). Coexistence in laboratory populations of *Paramecium aurelia* and its predator *Didnium nasutum*. *Ecology, 54,* 1320–1327.

Maes, P. (1990). *Designing autonomous agents, theory and practice from biology to engineering and back*. Cambridge, MA: MIT Press.

Marion G., Renshaw E., & Gibson G. (2000). Stochastic modeling of environmental variation for biological populations. *Theoretical Population Biology, 57,* 197–217.

Mirollo, R., & Strogatz, S. (1990). Synchronization of pulse-coupled biological oscillators. SIAM *Journal of Applied Mathematics, 50,* 1645–1662.

Morse, E. (1916, February 14). Fireflies flashing in unison. *Science, 43,* 169–170.

Murname, R., & Levy, F. (1996). *Teaching the new basic skills: Principles for educating children to thrive in a changing economy.* New York: Free Press.

Murray, J. D. (1989). *Mathematical biology.* Berlin, Germany: Springer.

Noss, R. (1988). The computer as a cultural influence in mathematical learning. *Educational Studies in Mathematics, 19*, 251–268.

Noss, R., & Hoyles, C. (1996). *Windows on mathematical meanings: Learning cultures and computers.* Dordrecht, The Netherlands: Kluwer.

Ogborn, J. (1999). Modeling clay for thinking and learning. In N. Roberts, W. Feurzeig, & B. Hunter (Eds.), *Computer modeling and simulation in science education* (pp. 5–37). Berlin, Germany: Springer-Verlag.

Papert, S. (1972). Teaching children to be mathematicians vs. teaching about mathematics. *International Journal of Mathematics Education in Science and Technology, 3,* 249–262.

Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. New York: Basic Books.

Popper, K. (1959). *The logic of scientific discovery.* London: Hutchinson.

Purves, W., Orians, G., & Heller, H. (1992). *Life: The science of biology* (3rd ed.). Sunderland, MA: Sinauer Associates.

Quine, W. V. (1960). *Word and object*. Cambridge, MA: MIT Press.

Reisman, K. (1998). *Distributed computer-based modeling*. Unpublished undergraduate honors thesis, Tufts University, Boston.

Repenning, A. (1993). *AgentSheets: A tool for building domain-oriented dynamic, visual environments*. Unpublished doctoral dissertation, University of Colorado, Boulder.

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

Resnick, M. (1994). *Turtles, termites and traffic jams: Explorations in massively parallel microworlds*. Cambridge, MA: MIT Press.

Resnick, M. (1996). Beyond the centralized mindset. *Journal of the Learning Sciences, 5,* 1–22.

Resnick, M., & Wilensky, U. (1993, April). *Beyond the deterministic, centralized mindsets: New thinking for new sciences*. Paper presented at the meeting of the American Educational Research Association, Atlanta, GA.

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

Roberts, N., & Barclay, T. (1988, Fall). Teaching model building to high school students: Theory and reality. *Journal of Computers in Mathematics and Science Teaching, 8*(1), 13–24.

Rosenzweig*,* M. (1971, January 29). Paradox of enrichment: Destabilization of exploitation ecosystems in ecological time. *Science, 171,* 385–387.

Salk, J. (1983). *Anatomy of reality: Merging of intuition and reason.* New York: Columbia University Press.

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

Smith, J. P., diSessa, A. A., & Roschelle, J. (1994). Reconceiving misconceptions: A constructivist analysis of knowledge in transition. *Journal of the Learning Sciences, 3,* 115–163.

Stroup, W. (1996). *Embodying a nominalist constructivism: Making graphical sense of learning the calculus of how much and how fast.* Unpublished doctoral dissertation, Harvard University, Cambridge, MA.

Taylor, C., Jefferson, D., Turner, S., & Goldman, S. (1989). RAM: Artificial life for the exploration of complex biological systems. In C. Langton (Ed.), *Artificial life* (pp. 275–295). Reading, MA: Addison-Wesley.

Tisue, S., & Wilensky, U. (2004, May). *NetLogo: A simple environment for modeling complexity.* Paper presented at the Fifth Proceedings of the International Conference on Complex Systems, Boston.

Volterra, V. (1926, October 16). Fluctuations in the abundance of a species considered mathematically. *Nature, 118,* 558–560.

White, B., & Horwitz, P. (1988). Computer microworlds and conceptual change: A new approach to science education. In P. Ramsden (Ed.), *Improving learning: New perspectives* (pp. 69–80). London: Kogan Page.

Wilensky, U. (1993). *Connected mathematics: Building concrete relationships with mathematical knowledge*. Unpublished doctoral dissertation, Massachusetts Institute of Technology, Cambridge.

Wilensky, U. (1995). Learning probability through building computational models. In *Proceedings of the Nineteenth International Conference on the Psychology of Mathematics Education* (pp. 152–159). Recife, Brazil: Program Committee of the 19th PME conference.

Wilensky, U. (1996). Modeling rugby: Kick first, generalize later? *International Journal of Computers for Mathematical Learning, 1,* 125–131.

Wilensky, U. (1997a). StarLogoT 2002 [Computer software]. Evanston, IL: Northwestern University, Center for Connected Learning and Computer-Based Modeling.

Wilensky, U. (1997b). What is normal anyway? Therapy for epistemological anxiety. *Educational Studies in Mathematics, 33,* 171–202.

Wilensky, U. (1998). Connected models [Computer software]. Evanston, IL: Northwestern University, Center for Connected Learning and Computer-Based Modeling.

Wilensky, U. (1999a). GasLab—An extensible modeling toolkit for exploring micro- and macro- views of gases. In N. Roberts, W. Feurzeig, & B. Hunter (Eds.), *Computer modeling and simulation in science education* (pp. 151–178). Berlin, Germany: Springer-Verlag.

Wilensky, U. (1999b). NetLogo (Version 3.0.2) [Computer software]. Evanston, IL: Northwestern University, Center for Connected Learning and Computer-Based Modeling.

Wilensky, U. (2001, August). *Modeling nature's emergent patterns with multi-agent languages*. Paper presented at Eurologo 2001, Linz, Austria.

Wilensky, U., & Resnick, M. (1995, April). *New thinking for new sciences: Constructionist approaches for exploring complexity.* Paper presented at the meeting of the American Educational Research Association, San Francisco.

Wilensky, U., & Resnick, M. (1999). Thinking in levels: A dynamic systems perspective to making sense of the world. *Journal of Science Education and Technology, 8,* 3–19.