

Preface

The Simulation Model: A Left-Brain Tool for Right-Brain Scientists

In the domain of ecology, there exists a huge source of information that is largely undocumented and therefore unavailable to practitioners. It is expertise that is sequestered in the individual minds of many field ecologists and rarely captured in a form that is readily accessible by the greater community of practice. The nature of this expertise differs depending on the interests and working style of the practitioner. Some ecologists seek documentable precision in knowledge by investigating natural systems through the collection of large data samples capable of producing statistically verifiable insights. This quantitative approach can offer intimate and accurate understandings of small subsets of an ecosystem. Other ecologists develop their knowledge by conducting diverse case studies designed to inform a larger overview. Both approaches lead ecologists to develop valuable insights on how ecosystem components function and interact. Each individual's growing expertise constitutes a part of a rich, but uncompiled, knowledge base. It is available to the possessor and associates for specific projects or applications, but it remains generally, if unintentionally, concealed from the greater community of practice.

Psychology informs us that people have two different modes of thinking, each of which roughly correlate to one brain hemisphere or the other. Right-brain thinking is considered to be more creative, intuitive, holistic, and spontaneous, while left-brain thinking is considered to be more methodical, logical, linear, and analytical. In terms of ecological research, the synthesis of big-picture results from many case studies represents a right-brain approach, and the development and analysis of large data samples represents a left-brain approach. But because there is little overlap in the two approaches, we often have to choose between keen but unverifiable intuition, on one hand, and hard but never-complete data on the other. And these differences pose an understanding gap between experts from the two different methodological approaches.

This gap may be illustrated by the following scenario. Over many years, a field ecologist develops deep, intuitive insight into an ecosystem that makes it possible

for him or her to forecast the consequences of proposed management actions on an ecosystem, often with a very high level of confidence. A planner who is considering new management initiatives may seek out the insights of the seasoned expert, whose reputation the planner knows and trusts. The field ecologist's expertise is often rooted more deeply in experience and intuition than in peer-reviewed research. If he or she wants to disseminate those insights to others beyond the immediate research team or work group, prospective users must be able to verify the validity and applicability of that expertise.

One approach the field ecologist can take to disseminate the use of hard-won technical insights is to apply left-brain skills to what is already understood intuitively—to explicitly identify and analyze the cause–effect relationships that lie beneath the intuitive knowledge. A computer simulation model is an excellent tool for capturing and representing such technical knowledge in a way that is highly explanatory and well documented. A simulation model can employ validated algorithms plus data and alternate assumptions to reflect the field ecologist's insight into the implications of environmental change or management actions. Simulation results can be compared with the ecologist's "instincts," both to assess the validity of the model and to further illuminate the right-brain thinking behind it. Any gaps revealed between simulation results and the ecologist's deep understanding can be considered and addressed. As the model is refined and simulation results match the right-brain understanding of the system, the ecologist achieves an analytical validation of ideas that may previously have been beyond the reach of the left brain. At that point, the model is ready to share and to apply to specific cases, which can help decision makers and the general public develop improved impact analyses and policy alternatives.

For more than 25 years, I have taught life science students at the University of Illinois at Urbana-Champaign how to simulate dynamic biological phenomena on computers. It is my favorite activity as a professor. Over the years, students have modeled a full gamut of biological activity, ranging from the disciplines of microbiology to genetic engineering, and covering the dynamics of the individual cell, bacteria, individual plants or animals, and large collections of organisms. I try to help them learn that the intent of building these models is to better understand function and limits for the ultimate purpose of informing good management practice.

Regardless of my enthusiasm and best efforts, I have not had unqualified success at teaching my students why I believe that dynamic modeling and the acquisition of systems thinking capabilities are so essential to their future work. Below, I explore why this has happened and what might be done about it. I also will clearly lay out the general benefits of modeling. Students do well in my course in part because it is tailored to minimize reliance on sophisticated mathematics and programming. We are fortunate that model-building computer environments such as STELLA (isee systems, Lebanon, NH) and NetLogo (Wilensky 1999) are now available to help students to quickly and easily capture and document their ideas about biological dynamics as computer simulation models. The models created using these tools enable my students to clearly explain to me, to their other professors, and to the professional community the structure and dynamics of those areas where their specific interests lie.

Although students are not actively discouraged from model building by their thesis supervisors, they are not actively encouraged to investigate it, either. Most of

my students have enrolled in my modeling courses more on their own volition than on someone's advice. A second inhibitory factor is that modeling must be practiced continuously in order to develop skills and internalize them. Because the typical students in master's or doctoral programs in this area are under high demand to perform laboratory and field experiments, they find little time or incentive to build models for the purpose of capturing their understandings of how systems work.

Ecology-oriented students are traditionally focused on hypothesis-driven case studies and huge data collection projects that allow them to draw statistical inferences about how their systems function. This approach to research rarely allows one to infer behavior at one level from behavior observed at a lower level, or at one location in a landscape behavior observed in another. It does not help students to formalize first-principle understanding of the cause-and-effect functioning of their systems.

I have often speculated why this is the case. Is it because they are not trained in simulation modeling at an earlier age, as are engineering students, for example? Or do these students imagine that modeling and simulation require skills that are beyond their reach? (The ease of modeling using new and evolving software environments could dispel that notion, given some introductory hands-on instruction.) Or do such students really so love nature that they simply seek the means to dwell within it through ecological fieldwork? I would argue that this love of nature might be significantly enriched by starting the journey with a set of hypotheses, followed by a modeling exercise that can verify and improve their understanding of their system. A model offers students a means for testing their assumptions and questions and for identifying the parameters that must be investigated and verified by lab or fieldwork. It also can help students understand which parameters are the most important and which can be reliably derived from the literature.

I begin each course by sharing the idea that education has been evolving since literacy was solely found within the monastery, through the time when we realized that numeracy was required to distinguish the importance of our assumptions, to the present, when we find it necessary to add systems thinking to the list. Systems thinking helps us to more accurately formulate pertinent questions about the phenomena that interest us. As with the acquisition of literacy and numeracy, skill in systems thinking improves with practice. And the level of practice increases with improved understanding of the power of systems thinking. This explanation leads to my discussion of the power of systems thinking, and how dynamic systems simulation on the computer provides the key to this power.

It helps us to understand that we all model the dynamics of the world around us. We instinctively know how to duck a stone thrown at us, we know how to safely cross a street in fast, heavy traffic, and how to hit a baseball. We do this by first formulating a mental model of the process and the probable consequences of various alternative actions. We evolve this model by our own trial and error and by observation of the actions of others. Given that we all routinely construct mental models, it should come as no surprise that we can increase the complexity and explanatory power of those models by extending them with computer power. The application of computers to our models of the world expands the reach of our mind in a similar way that the telescope and microscope extend the reach of our eyes.

When we try to extend our mental models exclusively through thought to solve complex social, political, or economic problems, for example, we encounter three specific difficulties. First is the uncertainty of our grasp of the important features of the problem; second is the effects of responses to our interventions or to internal forces driven by complex feedback loops; and third is the delay between the interventions (or forces) and the reactions to them. This uncertainty—these feedbacks and delays—can so complicate the dynamics of a system that the human minds cannot account for them all unaided. Society has reached the point where the complexity of environmental, interpersonal, and interagency connections is growing faster than the human mind can evolve to comprehend them. So instead of waiting for evolution, humans invent the means to extend our senses—and now, our capacity to apply logic—in order to master the complexities of the system in a timely way. To my mind, that is the great promise of simulation modeling technology.

But what are the specific benefits of computer-aided dynamic modeling? Over the years, with the help of many others, I have compiled a list of such benefits. Presented roughly in order of importance, dynamic modeling:

1. Can highlight the gaps in our understanding of the system processes. The construction of a computer model requires us to systematically lay out the stocks and flows within a system and to identify the nature of the systems controls. It helps us to establish a hierarchy of importance of system parameters. It enables us to identify and challenge the assumptions behind our understanding of the process. Simulation results, along with clear documentation of the model structure, make it possible to provide a common frame of reference for all those involved in studying and managing the system.
2. Provides a system memory. Model building is the process of formally building and joining models of the component parts of a system to create a published description of it. Every validated model iteration contributes to a more realistic model of the whole system for everyone who is interested.
3. Reveals “normal” system performance. Large changes in a system’s behavior are, many times, just rare events that a good system model would show to be expected and at what frequency. Managers of such systems, without the aid of a model, tend to implement changes based on the occurrence of these rare but potentially expectable events. Such management actions, if based on a misdiagnosis of the environmental stress, can produce delayed reactions that have the potential to throw the system into disarray.
4. Allows testing *what-if* scenarios and experimentation with various kinds and levels of system management. A dynamic model makes it possible to see what happens when a system fails without any real-world consequence, and at far lower cost than witnessing an actual failure of the real system.
5. Provides quantitative information about the system operation at organizational levels (e.g., landscape or biome) and time scales (e.g., centuries) not ever experienced by observers of the real system.
6. Reveals emergent properties of the system, such as reactions and new states anticipated by no one involved in the study of the system. In other words, a dynamic model makes it possible to develop realistic predictions of a complex system under dynamic conditions.

7. Allows for “mediated modeling,” which involves all appropriate experts and stakeholders, and facilitates the development of consensus in complex or controversial situations. Current software is user-friendly and transparent enough that novices can quickly understand that their views are being accurately captured in the model. Once this is accomplished for all of those involved, the simulation results are more credible and, therefore, more readily accepted by all. Mediated modeling also can shed light on the accuracy of contending theories about system functions.
8. Promotes the accurate formulation of novel, previously unanticipated questions about system performance.

If these benefits provide sufficient motivation for the student to begin the investigation and practice of model building, then it is appropriate to generally outline what is involved in the modeling process.

The most suitable environment for creating spatially explicit dynamic models will be simple to learn but capable of handling high complexity. It should serve as a stepping stone to compiled modeling languages such as C+ when the form of the model has become fixed and intensive parameter testing is required. The programming language should make maximum use of symbols for the state and control variables in order to take advantage of our ability to quickly understand such symbolism. The programming language itself should be capable of handling statements in English-like language and provide efficient input from data sources. The language should be capable of graphical data input and have some ability to model spatially. It should allow easy testing of the effects of parameter variation.

STELLA (<http://www.iseesystems.com/>) fully meets these requirements, so it is ideal for those who are beginning to model and wish to explore while easily changing model structure and controls. STELLA is a simulation modeling environment that allows one to graphically capture the cause–effect relationships of a system that affect state variables. Equations and logic are then added to determine rates of flows in the state variables during a predetermined time step. When the model is finished to the developer’s satisfaction and is ready for extensive parameter sensitivity testing, curve fitting the model results to known data, or optimizing a certain state variable, another program is needed. My students use Berkeley-Madonna (<http://www.berkeleymadonna.com/>) to transform a STELLA model into a compiled form that runs many times faster than it can natively in STELLA. The Berkeley-Madonna program (1) runs extensive parameter sensitivity trials, (2) fits the model results to a given set of data, and (3) optimizes a given state in the model. The second item treats the model as though it was a regression “equation,” allowing that equation to embody all of our specific understanding of the system.

STELLA is most useful for modeling systems that are homogeneous in space. If the dynamic system model requires specific location-dependent detail, one can develop the model for each cellular space (or cell) in STELLA, and then translate those into the NetLogo modeling environment (Wilensky 1999, <http://ccl.northwestern.edu/netlogo/>) to capture the spatial dynamic process. Each parameter is set using a digital map to represent its geographical variation.

The NetLogo environment is the best compromise between the simple programming requirements of STELLA (which is ideal for either a single-cell model or a spatial model with no more than, say, 25 cells) and the complex programming required to knit thousands of cellular models together into a dynamic whole. One can learn a significant amount from a STELLA model, and it is always useful to begin one's ecological modeling there. But the resulting model will need to be restated in NetLogo with added programming to incorporate the maps of the constants and initial state values. It is quite possible that the slightly more demanding programming skills needed for using NetLogo will eventually evolve into an even simpler procedure. Our practice is to use either free or commercially available software and concentrate on the process of modeling instead of developing a spatial modeling program of our own.

Having taught the spatial dynamic modeling course at the University of Illinois for more than 20 years, with the help of Dr. James Westervelt and Dr. Charles Ehlschlaeger, we have evolved what I believe to be the best current way to learn the process. We start the class by dividing the students into teams of two or three, with each team focusing on a specific set of modeling questions. The first 2 weeks are spent learning NetLogo, and the rest of the course is devoted to finishing the model, preparing the map data, and answering the modeling questions.

Some class projects have blossomed into large follow-on projects, including master's theses and doctoral dissertations. The LEAM urban development model (<http://www.leam.uiuc.edu/>) originated in this class and is now the basis of a company and a university laboratory. Our model of the Mississippi River aquatic ecosystem is another such project, having begun in the class and now the basis for a major interuniversity project. As these models matured and grew to the point of tens of millions of cells, the programs were rewritten in C++, which greatly accelerated simulation speed but required more esoteric knowledge to revise the model.

I cannot overstate to life science and social science students the importance of first formulating clear and concise questions about the phenomenon of interest. After that, they should construct a model—first in STELLA—of the part of the ecosystem that is most directly relevant to answering their questions about it. This two-phase process, if well executed, will reveal after relatively little time and expense the parameters to which the model is very sensitive. Discovering the values of these key parameters becomes the objective of their lab and field experiments. Data from the literature may be sufficient to obtain the rest of the parameters. This process reduces the overall research work and makes its progress more predictable.

Urbana, IL, USA

Bruce Hannon

Reference

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

Ecologist-Developed Spatially-Explicit Dynamic
Landscape Models

Westervelt, J.D.; Cohen, G.L. (Eds.)

2012, XX, 260 p. With online files/update., Hardcover

ISBN: 978-1-4614-1256-4